

Registro automático de imágenes médicas preoperatorias e intraoperatorias basado en intensidad: Aplicación a un sistema de navegación quirúrgica.

TESIS DE MAESTRÍA

PRESENTADO POR:

GUSTAVO ADOLFO OSPINA TORRES

DIRECTOR: M.SC WALTER SERNA SERNA

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

FACULTAD DE CIENCIAS BÁSICAS

MAESTRÍA EN INSTRUMENTACIÓN FÍSICA

PEREIRA 2015

Agradecimientos

Ante todo le agradezco a Dios por haberme guiado y dado fuerzas para seguir adelante, a mis padres por su amor y constante apoyo y a mis amigos del grupo de investigación Applied Neuroscience, en especial al ingeniero Walter Serna, que me dieron la oportunidad de participar en el proyecto de investigación.

La investigación relacionada a este documento fue ejecutada gracias al Instituto de Epilepsia y Parkinson del Eje Cafetero – NEUROCENTRO S.A y al Departamento Administrativo de Ciencia, Tecnología e Innovación Colciencias, por la convocatoria 642: Locomotora de la innovación para el apoyo del desarrollo tecnológico, en el proyecto Sistema de cirugía guiada por imágenes para procedimientos de cerebro y columna.

Índice general

	Página
Índice de figuras	5
Índice de cuadros	7
Introducción: Resumen y Objetivos	9
1. Marco teórico y estado del arte	12
1.1. Registro de imágenes	12
1.1.1. Algoritmos basados en intensidad.	14
1.1.2. Modelos de transformaciones geométricas o espaciales	19
1.1.3. Optimizadores	23
1.1.4. Fusión de imágenes	26
1.1.5. Técnicas para la evaluación del registro	28
1.2. Imágenes diagnósticas tipo DICOM tipos usados y características	29
1.2.1. Técnicas de imágenes diagnósticas	29
1.2.2. Tipos de imágenes usadas	29
1.2.3. El estándar DICOM	31
1.2.4. Estructura de un archivo DICOM	32
1.2.5. Comunicación entre dispositivos DICOM	33
1.3. Estado del arte	35

2. Diseño de los algoritmos: pruebas e implementación	40
2.1. Métricas	40
2.2. Algoritmos en Matlab	46
2.2.1. Optimizador	47
2.2.2. Información mutua	47
2.2.3. Histograma conjunto:	47
2.2.4. Transformación geométrica	47
2.3. Implementación como librería del neuronavegador	48
2.3.1. Preproceso en las imágenes adquiridas	51
2.3.2. Adquisición de imágenes	53
2.3.3. Comunicación DICOM	55
3. Validación, resultados y conclusiones	58
3.1. Validación del registro	58
3.1.1. Precisión	58
3.1.2. Confiabilidad y tiempo de computo	61
3.1.3. Robustez	67
3.2. Conclusiones, recomendaciones y trabajos futuros	73
3.2.1. Conclusiones	73
3.2.2. Recomendaciones	73
3.2.3. Trabajos futuros	74
A. Manual Librería algoritmos	75
A.0.4. Algoritmos Matlab	75
A.0.5. Funciones adicionales	77
A.0.6. Funciones en el lenguaje CSharp	79
B. Manuales sobre comunicación DICOM	85
B.1. Manual DVTK	85

B.1.1. Configuración del modulo Query Retrieve SCP Emulator	86
B.1.2. Configuración del módulo Store SCP emulator	88
B.1.3. Dicom Network Analyzer	89
B.2. Manual GDCM	89
B.2.1. Instalación	90
B.2.2. Programa gdcmscu	90
B.2.3. Librería de programación GDCM en Csharp	92
B.3. Manual servidor DICOM: ORTHANC	94
Bibliografía	98

Índice de figuras

1.1. Imágenes de CT y MRI: sin registrar y registradas	13
1.2. Histogramas conjuntos	15
1.3. Ángulos de Euler o de navegación	20
1.4. Algunas técnicas de transformación espacial	22
1.5. Pasos para la realización de un registro:	27
1.6. Neuronavegador creado por el grupo Applied Neuroscience	31
2.1. Imagen genérica para pruebas	40
2.2. Muestra de 57×41	41
2.3. Imágenes de rodilla con MRI diferentes secuencias: izquierda con eco de spin con recuperación invertida , derecha: con eco de spin	45
2.4. Set de Imágenes monomodal	50
2.5. Ejemplo de estudios provenientes de TAC (a) y RM(b)	54
2.6. Ejemplo de estudios provenientes de US (a) y Ax(b) sin editar	54
2.7. Ejemplo de estudios provenientes de US (a) y Ax(b) editados	55
2.8. Diseño de la etapa de adquisición y registro con el angiógrafo	56
2.9. Diseño de la etapa de adquisición con equipos de ultrasonido	56
3.1. Phantom utilizado por el Grupo Applied Neuroscience	59
3.2. Imágenes del Phantom en TAC (a) y Ax (b)	59

3.3. Distancias y puntos tomados en el Phantom, a: Distancias tomadas en el registro y fusión de las imágenes de angiografía y TAC. b: Puntos tomados en el Phantom en la imagen registrada de 3.2b	60
3.4. Ejemplo de registro y fusion de una imagen de angiografia y una de TAC provenientes del phantom de la figura 3.1,	62
3.5. Set de imágenes multimodales fusionadas pero no registradas de la figura 2.3	63
3.6. Registro usando: a) Comando imregister matlab transformación rígida. b) Registro del algoritmo diseñado con transformación rígida	63
3.7. Registro usando: a) Comando imregister matlab transformación de similitud. b) Registro del algoritmo diseñado con transformación de similitud	64
3.8. Registro y fusión de las imágenes 3.5: la imagen fue previamente escalada e inclinada	64
3.9. Variación de IM (a) y la IMN(b), con imagen monomodal respecto al ruido gaussiano con varianza=0.01 ,0.02,0.03.	68
3.10. Variación de IM (a) y la IMN(b) con imagen multimodal respecto al ruido gaussiano con varianza=0.01 ,0.02,0.03. en v=0.03 la IMN falla	69
3.11. Variación de IM (a) y IMN (b) con filtros reductores de ruido	70
3.12. Variación de la IM respecto a la corrección gamma	70
3.13. Variación de la IMN respecto a la corrección gamma	71
 B.1. Modelo de informacion de QR SCP emulator	 87
B.2. Configuración de QR SCP emulator	87
B.3. Prueba de conexión entre GDCM y QR SCP emulator	88
B.4. Configuración del Store SCP emulator	88
B.5. Prueba del comando C-FIND y C-MOVE	92
B.6. Aplicacion de prueba creada en Csharp	95
B.7. Servicio Orthanc en windows	96
B.8. Servicio WEB Orthanc en windows	96
B.9. Query/Retrive (C-FIND)	97

Índice de cuadros

2.1. Tabla norma L1: valor del resultado de la norma L1 y si esta tiene éxito al encontrar al encontrar la muestra	42
2.2. Tabla SSD:valor del resultado de la SSD y si esta tiene éxito al encontrar al encontrar la muestra	42
2.3. Tabla CC:valor del resultado del CC y si este tiene éxito al encontrar al encontrar la muestra, con su error porcentual	42
2.4. Tabla del radio mínimo: valor del resultado del radio mínimo y si este tiene éxito al encontrar al encontrar la muestra, con su error porcentual	43
2.5. Información mutua de Shannon:valor del resultado de la IM y si esta tiene éxito al encontrar al encontrar la muestra, con su error porcentual	43
2.6. Información mutua normalizada:valor del resultado de la IMN y si esta tiene éxito al encontrar al encontrar la muestra, con su error porcentual	44
2.7. Información mutua de Rényi con α 2: valor del resultado IMR y si esta tiene éxito al encontrar al encontrar la muestra, con su error porcentual	44
2.8. Interpoladores usados: cúbico o bicúbico, Lineal, Vecinos mas cercanos, Lancos4	49
2.9. Tiempo de computo Algoritmos de optimización PSOAIW, PSOCIW, PSOLDW,DE,CS . .	51
2.10. Algoritmo PSOAIW	52
2.11. Optimización con Información mutua normalizada	53
3.1. Precisión del algoritmo con un imagen de TAC con Angiografía: D1=distancia 1, D2= distancia 2, D3= distancia 3, Angulo	60
3.2. Precisión del algoritmo con un imagen de TAC con Angiografía: Diferencias en las coordenadas X,Y de los puntos 1 y 2	61
3.3. Precisión del algoritmo con un imagen de TAC con Angiografía: Diferencias en las coordenadas X,Y de los puntos 3 y 4	62

3.4. Confiabilidad: set TAC, angiografía con una confiabilidad C=93 % usando IMN, optimizador: PSOAIW con epsilon =1.7E-4 e interpolador:cúbico, Se observó si tiene éxito o nó en el registro.	65
3.5. Confiabilidad C y tiempo de computo sets 2 y 3, se observo si tiene éxito o no en el registro.	66
3.6. tiempo de cómputo, sets 1,2,3 en el Neuronavegador	67
3.7. Cambio de la IM y la IMN respecto al ruido	68
3.8. Cambio de la IM y la IMN respecto al ruido en una imagen multimodal	69
3.9. Cambio de la IM y la IMN respecto a los reductores de ruido	69
3.10. Cambio de la IM y la IMN respecto a la corrección gamma	71
3.11. Cambio de la IM y la IMN respecto a la ecualización del contraste usando HE, BBHE, DISHE, CLAHE	72

Introducción

Resumen y Objetivos

Las cirugías guiadas por imágenes médicas han cobrado gran importancia en las última décadas y con ella los métodos de registro de imágenes, pues con esto se mejora la planeación, intervención y posterior análisis de ciertas cirugías que requieren una especial precisión como es el caso de las neurocirugías. Registrar un par o varias imágenes consiste en alinearlas de tal manera que una de las imágenes quede orientada de la misma forma de otra imagen que se toma de referencia, con esto se puede combinar la información de ambas en un mismo espacio y en una misma representación visual. El poder registrar las imágenes extraída antes de la cirugías (preoperatorias) con las imágenes hechas durante la cirugía (intraoperatorias) es de vital importancia para el especialista, pues puede observar los cambios que se realizan durante la cirugía y así actualizar la información anatómica y estructural, como por ejemplo la ubicación de un tumor el cual al realizar la cirugía cambia un poco en su posición o se deforma.

El objetivo primordial de esta tesis fue encontrar unas técnicas iniciales adecuadas que permitieron el registro automático de imágenes médicas provenientes de varias técnicas de creación imágenes y así se creo una librería computacional (prototipo) que pueda servir de base para crear un módulo de registro en un sistema de neuronavegación. Para cumplir con dicho fin se ha optado por usar tecnologías de similitud de imágenes basadas en intensidad como base principal para la realización del registro.

Como objetivos específicos se realizó primero un estudio para establecer las condiciones básicas de tratamiento, manipulación y pre-proceso si es necesario en las imágenes a registrar, luego se creó un algoritmo que realizó: una métrica de similitud de intensidades para determinar el grado de correspondencia de las imágenes a registrar, una parte encargada de la transformación espacial de las imágenes, con el fin de alinearlas, y una parte de optimización encargada de encontrar la mejor solución dado las coordenadas de transformación espacial. Por último se implementaron dichos algoritmos como una librería en el lenguaje C#, validando los algoritmos con la ayuda de un phantom y usando técnicas analíticas y visuales que permitió determinar la precisión, confiabilidad, robustez del algoritmo. Este estudio sirve de base para el posterior diseño de un módulo de registro para un sistema de neuronavegación.

Sistemas de cirugías guiados por imágenes han tenido un gran desarrollo en las últimas décadas pues ayudan al cirujano a guiarse dentro de un espacio físico y y permiten realizar operaciones mucho más precisas y así realizar una mejor planeación y obtener mejor desempeño en la cirugía. Sin embargo un gran problema en las operaciones cerebrales o neurocirugías, como en las de la espina

dorsal, es como el cirujano pueda combinar la información preoperatoria (antes de la cirugía) del paciente con información intraoperatoria (durante el procedimiento); pues la información preoperatoria puede quedar rápidamente desactualizada y es necesario actualizarla. Dichos inconvenientes pueden llevar a complicaciones como es el caso de problemas asociadas con la mala colocación de tornillos pediculares en las operaciones de fusión espinal como se afirma en [1], donde se ve una tasa importante de colocaciones incorrectas de los tornillos pediculares. Otro caso sería el brain shift [2], en el cual una estructura se desplaza al realizar el procedimiento y por lo cual la información preoperatoria queda desactualizada.

Los sistemas de neuronavegación pretenden dar solución a estos inconvenientes pues es un sistema basado en la localización intraoperatoria guiada por imágenes sin la ayuda de un marco estereotáxico, el cual cumple con la función de definir la posición de un punto dado en el campo quirúrgico. El neuronavegador consiste en un método de registro de la imagen diagnóstica y el espacio físico, un instrumento de localización y una visualización de las imágenes médicas obtenidas y procesadas por un ordenador, proporcionando una retroalimentación y visualización en tiempo real de la localización de los instrumentos quirúrgicos. Aunque la información suministrada por el neuronavegador es de vital importancia para este tipo de operaciones, las imágenes con el cual se desarrolla el modelo virtual del paciente son preoperatorias, o sea que se tomaron antes de realizar la operación. Una vez la operación quirúrgica esté en marcha muchos cambios pueden darse en el paciente, haciendo que el neuronavegador pierda su referencia espacial. Debido a esto se suele acompañar a estas cirugías de ciertas tecnologías como el ultrasonido y el fluoroscopio, con lo cual se puede visualizar en el acto la estructura corporal. Sin embargo el especialista debe formar una percepción tridimensional del espacio a partir de la proyección bidimensional entregada por el fluoroscopio o ultrasonido si no cuentan con visualización 3D, lo que reduce considerablemente la percepción del usuario. Además en el caso del fluoroscopio exponer tanto al paciente como a los que se encuentren en la sala a una dosis de rayos X.

Según la Organización Mundial de la Salud en [3, 4] entre el 8 % de la población mundial padece de alguna clase de desorden neurológico, siendo un 7 % para el continente americano. Mientras que a nivel mundial se registra que un 11.7 % de las muertes están asociadas a estos desórdenes y más del 80 % de estas muertes se producen en países de ingresos bajos o medianos. Aunque no es la principal causa de muerte, se evidencia que los desórdenes neurológicos son los que más reducen la calidad de vida tanto de los pacientes como de las personas que los rodean. También son los que más incapacitan a la población, por lo que hoy en día se le considera un problema de salud pública.

Entre los trastornos neurológicos tenemos la epilepsia, la enfermedad de Alzheimer y otras demencias, enfermedades cerebro vasculares tales como los accidentes cerebro vasculares, la migraña y otras cefalalgias, la esclerosis múltiple, la enfermedad de Parkinson, las infecciones neurológicas, los tumores cerebrales, las afecciones traumáticas del sistema nervioso tales como los traumatismos craneoencefálicos, y los trastornos neurológicos causado por la desnutrición como se evidencia en [3].

En el informe de la OMS sobre desórdenes neurológicos [5] se cita que los costos provocados por estas enfermedades por ejemplo en Europa ascienden a 139.000 millones de euros, por lo tanto en países de bajos o medios recursos les será más complicado afrontar los tratamientos necesarios para estos trastornos y muchas personas afectadas por trastornos neurológicos, o quienes los atienden o sus familias, tienen dificultades para acceder a los cuidados apropiados, por lo tanto la OMS aboga por que la atención en neurológica se integre en la atención primaria en salud.

En Colombia como se ve en el atlas [6]: Country resources for neurological disorders hecho

por la OMS el número de neurocirujanos está entre 0.1 a 1 por cada 100000 habitantes, en las Américas tiene una media de 0.76 %, lo cual indica que en Colombia está en una media aceptable y por lo tanto el desarrollo o mejoras de estas tecnologías es de gran importancia para darle un mejor atención al paciente y tenga más posibilidades de recuperación.

También hay que tener en cuenta que en la mayoría se los procedimientos en neurocirugías como en columna requieren una precisa localización del objetivo que se desea alcanzar como: tumores, malformaciones, inserción de herramientas y dispositivos etc y así poder minimizar el daño a estructuras normales como lo menciona en [7]. Algoritmos de registro son críticos en muchas partes de estos procedimientos como por ejemplo en la identificación pre quirúrgicas de anomalías, planeamiento, cirugías guiadas por imágenes intraoperatorias y evaluaciones postoperatorias. Convencionalmente cirugías guiadas por imágenes son hechas por cirujanos usando su percepción visual y examinando imágenes 2D provenientes de MRI o CT y mentalmente relacionan estas imágenes pre operativas con la información que obtienen del paciente. Este forma de proceder es muy dependiente de la experiencia del cirujano, por lo tanto el registro automático de imágenes puede asistir al cirujano para guiarse en estos procedimientos y así obtener una precisa fusión de imágenes preoperatorias con intraoperatorias. Otro aspecto importante es que aunque los sistemas de neuronavegación actuales ya incluyen métodos para el registro de imágenes preoperatorias e intraoperatorias como es el caso de los sistemas creados por Brainlab <https://www.brainlab.com/en/surgery-products/overview-spinal-trauma-products/image-registration/> y Medtronic <http://superdimension.com/products/superdimension-system/features/>, los cuales son caros y por lo tanto privativos en regiones de escasos o medios recursos como Colombia por lo tanto sistemas de neuronavegación como los que vienen desarrollando por Instituto de Epilepsia y Parkinson del Eje Cafetero en el grupo de investigación Applied Neuroscience vuelve este tipo de tecnologías más accesibles para cualquier entidad que la requiera.

Capítulo 1

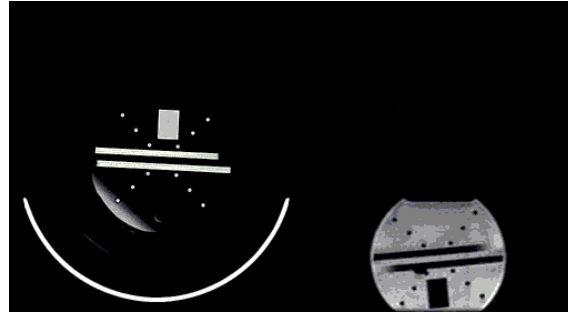
Marco teórico y estado del arte

El registro de imágenes es una importante herramienta en el campo de las imágenes médicas. En muchas situaciones varios tipos de imágenes son analizadas para dar un diagnóstico de la situación del paciente. Estas imágenes son adquiridas por diferentes tipos de dispositivos tales como: escáneres de rayos X, resonancia magnética (MRI), tomografía computarizada (CT), ultrasonido, etc. Cada uno de estos sensores puede dar diferente información del paciente tales como estructura corporal o dar idea de la función de estas. El registro de imágenes permite combinar la información de estas imágenes dadas por diferentes sensores y poder visualizarlas en una única imagen con el objetivo de facilitar el examen médico [8]. Además en sistemas de navegación quirúrgica guiada por imágenes el registro de imágenes es una gran herramienta pues permite al especialista ver como las estructuras corporales cambian debido a la intervención en el paciente y así tener una mejor perspectiva y precisión dando solución a problemas como el brain shift donde las estructuras cerebrales se deforman o se mueven debido a la intervención realizada pues se pierde liquido cerebro raquídeo [2].

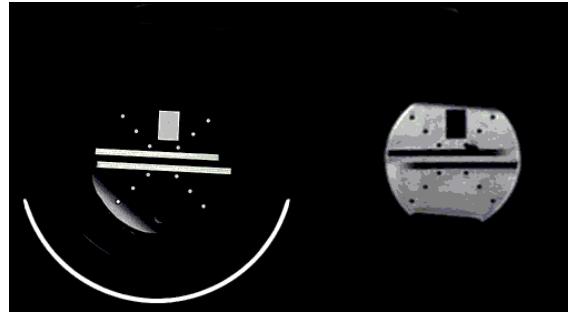
1.1. Registro de imágenes

El registro de imágenes es un proceso para determinar correspondencias entre dos o más imágenes, de este modo es posible combinar información de imágenes provenientes de varias fuentes como resonancia magnética, ultrasonido, rayos X, etc y así es posible combinar en una misma imagen fusionada las características propias proveniente de varios sensores (multimodal) como se ve en la figuras 1.1a y 1.1b o también de un mismo sensor (monomodal) en donde pueden variar puntos de vista o lapsos de tiempo diferentes. Para esto se necesita una función de transformación que mapee las diferentes imágenes y así poder alinear las imágenes. El registro de imágenes también puede ser sobre un volumen de datos y así combinar información 3D con 2D descrito en [9].

Según [10], las técnicas sobre registro de imágenes se pueden dividir en dos grupos: basados en características, donde se extrae vía segmentación de la imagen algún tipo de característica saliente como puntos, líneas, bordes, etc y luego se comparan estas características; sin embargo se debe tener en cuenta que es susceptible a errores. Luego están las técnicas basadas en intensidad donde se compara los valores de voxels y pixels usando una medida que determine que tan parecidas



(a) Imágenes CT y MRI sin registrar



(b) Imágenes CT y MRI registradas

Figura 1.1: Imágenes de CT y MRI: sin registrar y registradas

son las imágenes la cual está asociada por la estadística de la imagen, es típicamente más lenta pero no requiere segmentación.

Se pueden definir 3 factores claves en el registro de imágenes: la imagen de referencia o fija la cual permanece sin cambios en el registro, la imagen móvil o objetivo la cual cambia para adecuarse a la imagen de referencia y la transformación geométrica que es necesaria hacer para registrar la imagen objetivo a la fija .

Para el registro de dos imágenes una de las imágenes, la imagen móvil u objetivo $I_m(x)$, es deformada para ajustarse a la imagen referencia $I_f(x)$. Las dos imágenes son de dimensión d y cada una es definida en su propio dominio espacial respectivamente. El registro consiste en buscar un desplazamiento u_x que haga que $I_m(x)$ se alinee espacialmente a I_f . Una formulación diferente dice que el problema del registro de imágenes es encontrar una transformación $T(x)$ que haga que I_m se alinee con I_f . La transformación es definida como un mapeo desde la imagen de referencia a la objetivo [8].

Para la realización exitosa del registro son necesarios los siguientes pasos según [9]:

- **Pre-proceso:** esto involucra el procesamiento previo de las imágenes para facilitar la extracción de características y su correspondencia. Usando métodos tales como ajuste de escala, remoción de ruido y segmentación. Se ajusta la escala de una de las imágenes hacia la otra haciendo un remuestreo y así facilitando la extracción de características. Si la imagen tiene mucho ruido se suavizan para reducirlo. La segmentación divide la imagen en regiones donde

se extraen las características.

- **Selección de características:** para el registro de dos imágenes, un número de características son seleccionadas y se establece una correspondencia entre ambas. Conocidas las correspondencias, una función de transformación es establecida para remuestrear la imagen objetivo a la imagen de referencia. Algunas características usadas son esquinas, líneas, curvas, contornos, o en general alguna forma específica. También se puede optar por usar como característica las intensidades de los píxeles o voxels lo cual resultaría en un método estadístico. El tipo de característica dependerá del tipo de imágenes, por ejemplo: una imagen hecha por el hombre suele tener muchos segmentos de líneas, mientras que una por satélites presenta regiones o contornos. del algoritmos
- **Correspondencia de características:** esto puede ser conseguido seleccionando características en la imagen de referencia y buscándolas en la imagen objetivo o seleccionando características en ambas imágenes y determinar una correspondencia entre ellas. También se pueden comparar estadísticamente las imágenes y mirar la similitud entre las dos imágenes. En esta etapa se puede incluir una fase de optimización .
- **Determinación de la función de transformación:** conociendo las coordenadas de un conjunto de correspondencias entre las imágenes, se determina una función de transformación geométrica como rotación, traslación, escala, entre otras para remuestrear la imagen objetivo a la geometría de la imagen de referencia. Se puede optar por una transformación rígida y no rígida.
- **Fusión y remuestreo:** conocida la función de transformación se aplica la imagen objetivo la cual determinada dicha transformación puede moverse rotar e incluso cambiar de escala, entre otras y se remuestrea a la de referencia; se puede realizar la fusión de las dos imágenes y combinar así sus informaciones y determinar cambios en las escenas.

Para la realización de este proyecto se ha optado por solo usar algoritmos estadísticos basados en intensidad pues estos suelen ser automáticos [10].

1.1.1. Algoritmos basados en intensidad.

Algoritmos basados en intensidad, también llamados medida de la similitud de voxels, dominan el campo para el registro de imágenes médicas multimodales como se aprecia en [11]. Para esto es necesario tener métrica o función de costo basado en intensidad que examine las dos imágenes a través de la transformación geométrica dada y buscar donde es máxima o mínima esta función. Es útil usar alguna función de optimización para encontrar algún valor inicial próximo al cual se desea transformar. Cuando el registro se realiza entre imágenes multimodales, o sea provenientes de diferentes sensores, la intensidad de los voxels en ocasiones es muy diferente, así que técnicas que comparen directamente la intensidad de los voxels son inútiles y es necesario recurrir a la teoría de la información para comparar la información común entre éstas como es el caso de la técnica llamada información mutua, basada en entropías de la información. Si las imágenes fueron tomadas por el mismo sensor o con una descripción de voxels parecida se pueden utilizar técnicas monomodales, las cuales comparar directamente la intensidad de los voxels y requieren menos tiempo de cómputo.

En muchas de las técnicas que se verán a continuación es necesario sacar la densidad de probabilidades de las imágenes a registrar, lo cual indica sacar el histograma a la imagen y dividir los

niveles por el número total de píxeles en la imagen.

Densidad de probabilidades e Histogramas y conjuntos:

La relación entre intensidades de dos imágenes es reflejada en la distribución de probabilidades conjuntos (DPC). Para obtener el DPC de un par de imágenes es necesario calcular el histograma conjunto y dividir cada valor del histograma entre el número total de píxeles en la imagen como se ve en [12]. Las dos imágenes deben tener igual tamaño y tener la misma profundidad de colores.

Los histogramas conjuntos es una herramienta usada para la visualización de la relación entre correspondencia de intensidades de una o más imágenes como se aprecia en las figuras 1.2a y 1.2b. Para dos imágenes A y B, el histograma es de 2 dimensiones y es construido graficando la intensidad a de cada voxel en la imagen A contra la intensidad b de cada voxel en la imagen B. El valor de cada locación en el histograma $h(a,b)$ corresponderá a el número de voxels con intensidad a en modalidad A y intensidad b en modalidad B como se ve en [11]. Para realizar el histograma conjunto es necesario que las imágenes sean de la misma dimensión. El tamaño de histograma h está determinado por el número de diferentes voxels en la imagen, por ejemplo: si la intensidad del pixel en A y B varía entre 0-255 (8 bits) entonces el histograma h tendrá una dimensión de 256×256 .

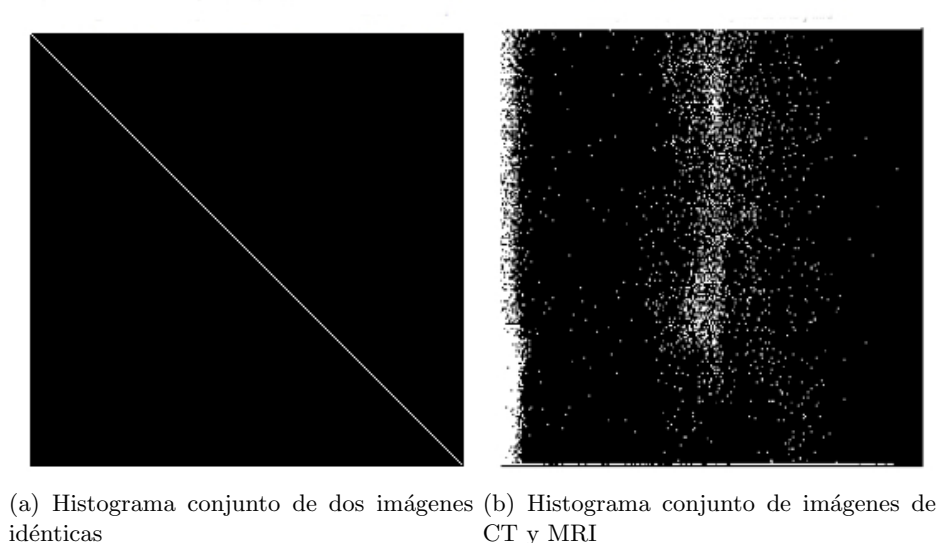


Figura 1.2: Histogramas conjuntos

Métricas y medidas de similitud o disimilitud

Una métrica es una función que examina la similitud o disimilitud entre dos imágenes, dependiendo del tipo de imágenes a registrar unas técnicas dan mejores resultados que otras. Si se asume que las imágenes a registrar son parecidas se pueden utilizar los siguientes métodos:

- **Suma de las diferencias absolutas , norma L1 o norma Manhattan:** es una de más antiguas medidas de disimilitud para comparar dos imágenes. Consiste en examinar el valor absoluto de las diferencias entre píxeles. La norma L1 entre dos imágenes está definida por la ecuación 1.1.

$$L_1 = \sum |A(i) - B(i)| \quad (1.1)$$

Si X y Y son un par de imágenes obtenidas por el mismo sensor para las mismas condiciones, y si el sensor tiene una alta relación señal/ruido esta medida puede obtener tan buenos resultados como otras técnicas más complejas como se ve en [12], pero tiene un pobre desempeño para imágenes multimodales.

- **Suma del cuadrado de las diferencias (SCD o SSD sum of square differences):** la medida de similitud más simple, consiste en mirar la diferencia entre los voxels y elevarlas al cuadrado como se ve en [11, 8, 7]. Para N voxels i en las imágenes A y B la SSD es: 1.2

$$SSD = \frac{1}{N} \sum |A(i) - B(i)|^2 \quad (1.2)$$

B es la imagen objetivo la cual se le aplica una transformación T. Este tipo de métrica funciona con imágenes muy similares que difieren muy poco la una de la otra, por lo tanto es una técnica aplicable solo para imágenes monomodales, solo es óptima cuando las dos imágenes solo difieren por el ruido gaussiano según [11].

- **Coefficiente de correlación de Pearson normalizado y no normalizado (CCN o NCC normalised correlation Coefficient):** para un caso menos estricto que la SSD se considerara una relación lineal entre los valores de las intensidad. La medida de similitud optima es el coeficiente de correlación que está definida en 1.3 como se ve en [11, 8, 7]

$$CC = \frac{1}{N} \frac{\sum_i (A(i) - \bar{A}) \cdot (B(i) - \bar{B})}{\{\sum_i (A(i) - \bar{A})^2 \cdot \sum_i (B(i) - \bar{B})^2\}^{\frac{1}{2}}} \quad (1.3)$$

Donde \bar{A} y \bar{B} son la media de la intensidad de los píxeles en A y B, la versión normalizada esta dada por

$$C = \frac{1}{N} \sum_i A(i) - B(i) \quad (1.4)$$

El coeficiente de correlación fue descubierto por Bravais en 1846 y luego Pearson demostró que es la mejor correlación entre dos secuencias de números. El Coeficiente de correlación r varía entre -1 y +1. Cuando es 1 existe una correlación perfecta entre las secuencias, cuando es -1 una secuencia es la negativa de la otra, de lo contrario las secuencias solo estarán relacionadas en parte. En el caso de imágenes si el coeficiente es 1 las dos imágenes son exactamente iguales como se ve en [12]

La suposición de la relación lineal entre los valores de los píxeles o voxels es por lo general poco realista para imágenes multimodales pero se usó para este fin en los primeros trabajos al respecto según [11]. Si X y Y representan intensidades en dos imágenes obtenidas bajo diferentes condiciones de iluminación y las intensidades están linealmente relacionada, existirá una gran similaridad entre las imágenes. Cuando estas son de diferente modalidad las relaciones no son lineales y por lo tanto no producirán un coeficiente de correlación alto y puede ocasionar errores. Esta técnica es relativamente eficiente pues requiere un número

pequeño de adiciones y multiplicaciones en cada pixel con lo cual puede reducir el tiempo de procesamiento.

- **Radio mínimo:** es una medida de similitud entre dos imágenes, si la imagen Y es una versión con ruido de la imagen X y si el ruido presente en Y es proporcional a la intensidad de la señal. El radio mínimo está definido por la ecuación 1.5 donde $r_i = \min(Y_i/X_i, X_i/Y_i)$,

$$m_r = \frac{1}{n} \sum_{i=1}^n r_i \quad (1.5)$$

El radio m_r será igual a 1 si las imágenes son iguales y decrece cuando X y Y no tienen mucha dependencia una de la otra. Como el ruido que varía con la intensidad tiene un efecto relativamente pequeño, la medida del radio m_r es muy cercano a 1. Este fenómeno puede aplicarse para el registro de imágenes donde hay cierta interferencia de ruido, pero como se ve en [12] es muy sensible a las diferencias entre intensidades, lo cual hace que no sea adecuado su uso para imágenes capturadas con diferentes condiciones de iluminación o de diferentes sensores (multimodales). La computación del radio mínimo es relativamente pequeña ya que tiene pocas operaciones.

- **Información mutua normalizada y no normalizada (IM o MI mutual información):** está basado en la teoría de la información [13]. Esta medida de similitud es la más usada para imágenes multimodales buscando información mutua que posean las imágenes según [11]. La técnica consiste en minimizar la entropía conjunta de las imágenes y por lo tanto maximizando su información mutua. Se define la IM en 1.6

$$I(A, B) = H(A) + H(B) - H(A, B) \quad (1.6)$$

Donde $H(A)$ y $H(B)$, es la entropía de Shannon (en honor a Claude E. Shannon) de la imagen A y B respectivamente y $H(A, B)$ es la entropía conjunta de las imágenes definidas 1.7, 1.8 y 1.9 respectivamente.

$$H(A) = - \sum_{a \in A} p_A(a) \log p_A(a) \quad (1.7)$$

$$H(B) = - \sum_{b \in B} p_B(b) \log p_B(b) \quad (1.8)$$

$$H(A, B) = - \sum_{a \in A} \sum_{b \in B} p_{AB}(a, b) \log p_{AB}(a, b) \quad (1.9)$$

Donde $H(a)$ o $H(b)$ es la medida de la incertidumbre promedio de una variable aleatoria a o b, $P_A(a)$ y $P_B(b)$ en este caso sería la probabilidad de encontrar un voxel o pixel en la imagen y $P_{AB}(a, b)$ es la probabilidad conjunta, para este caso sería el histograma conjunto dividido por el número total de píxeles en la imagen como se ve en [11], por lo tanto $H(A, B)$ es la entropía condicional. Si el logaritmo está en base 2 la entropía es representada en bits, si es logaritmo natural sería la entropía física según [14] [15].

La información mutua de Shannon fue utilizada para el registro de imágenes, basado en la observación que la densidad de probabilidades conjuntas de imágenes registradas era menos dispersa que el de imágenes sin registrar como se ve en las figuras 1.2a y 1.2b. La información mutua fue así usada para medir la dispersión de la densidad de probabilidades conjuntas,

cuando esta dispersión es mínima la dependencia de las intensidades de los pixeles es máxima como se ve en [12].

En el caso de la IM normalizada se puede obtener la ecuación 1.12 de 1.10 y 1.11.

$$\tilde{I}(A, B) = \frac{2I(A, B)}{H(A) + H(B)} \quad (1.10)$$

$$\tilde{I}(A, B) = H(A, B) - I(A, B) \quad (1.11)$$

$$\tilde{I}(A, B) = \frac{H(A) + H(B)}{H(A, B)} \quad (1.12)$$

Alternativas a la entropía de Shannon: se puede usar definiciones alternativas a la IM usando:

- **Información mutua usando entropía de Rényi:** Esta se define en termino de la entropía de Rényi creada por Alfréd Rényi que esta definida en 1.13 para una distribución discreta finita de probabilidades ver [12]-

$$E_\alpha = \frac{1}{1-\alpha} \log \left(\sum_i p_i^\alpha \right) \quad (1.13)$$

La entropía de Rényi de orden α es una generalización de la entropía de Shannon, ver [12], en donde si α es 1 se aproxima a la entropía de Shannon. La ecuación muestra la información mutua basada en el trabajo de Studholme [12], donde E_α^i y E_α^j son las entropías marginales y E_α^{ij} es la entropía conjunta de un par de imágenes.

$$R_\alpha = \frac{E_\alpha^i + E_\alpha^j}{E_\alpha^{ij}} \quad (1.14)$$

Como el orden α de la información mutua de Rényi se incrementa, las entradas en la densidades de probabilidades conjunta con altos valores serán magnificados, reduciendo el efecto de valores atípicos aleatorios en la densidad de probabilidades conjunta, esto genera que bajo ruido impulsivo o oclusiones, la información mutua de Rényi será mejor que la información mutua de Shannon, incluso con cero ruido medio, la de Rényi es mucho mejor que la de Shannon pero no por mucho. Computacionalmente la información mutua de Rényi es alrededor de 20 % a 30 % mas costosa que la de Shannon según [12].

- **Información mutua usando entropía de Tsallis:** Se define en términos de la entropía de Tsallis de orden q creada por Constantino Tsallis. Donde q es un numero real, como se ve en [14] y [15] y esta definida en la ecuación 1.15 cuando q se aproxima a 1 la entropía de Tsallis se aproxima a la de Shannon. La información mutua de Tsallis se puede definir en la ecuación 1.16

$$S_q(p_i) = \frac{k}{q-1} \left(1 - \sum_i p_i^q \right) \quad (1.15)$$

$$R_q = S_q^i + S_q^j + (1-q)S_q^i S_q^j - S_q \quad (1.16)$$

Donde S_q^i y S_q^j están definidos por 1.17 y 1.18 respectivamente

$$S_q^i = \frac{1}{q-1} \sum_{i=0} p_{ij} (1 - p_{ij}^{q-1}) \quad (1.17)$$

$$S_q^j = \frac{1}{q-1} \sum_{j=0} p_{ij} (1 - p_{ij}^{q-1}) \quad (1.18)$$

La entropía de Tsallis hace los valores atípicos menos importantes que en la entropía de Rényi cuando q toma un valor mas grande que 1, esto hace que la entropía de Tsallis sea menos sensitiva al ruido que la de Rényi y mas robusta que la de Shannon. Según [12] computacionalmente la información mutua de Tsallis es mas costosa que la de Rényi porque reemplaza una evaluación de un logaritmo con un numero de multiplicaciones.

Existen más algoritmos pero uno de los más usados para imágenes multimodales es el basado en la información mutua.

1.1.2. Modelos de transformaciones geométricas o espaciales

Las transformaciones geométricas juegan un papel central en cualquier procedimiento de registro de imágenes. Estos modelos imponen una distorsión geométrica que se aplica durante el proceso de registro. Existen dos tipos de transformaciones rígidas y no rígidas, las rígidas debido a su naturaleza lineal preservan todas las distancias y ángulos internos, las no rígidas pueden deformar la imagen en estos casos es necesario usar un interpolador como se ve en [16, 8, 17].

Cualquier sistema de transformación lineal en dos dimensiones puede expresarse como una matriz 3×3 y una de 3 dimensiones puede expresarse como una matriz de 4×4 . En 2D cualquier coordenada (x,y) se el puede asignar a un vector 1.19

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1.19)$$

Y en 3D un punto (x,y,z) puede asignarse al vector 1.20 , el último término puede verse como parámetro para que las matemáticas relacionadas funcionen perfectamente

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (1.20)$$

Entre las transformaciones geométricas tenemos:

- **Traslación:** consiste simplemente en trasladar la imagen en alguna coordenada cierto vector t ver figura 1.4 y está definida por 1.21

$$T_{\mu}(x) = x + t \quad (1.21)$$

Para el caso de coordenadas cartesianas de dos dimensiones:

$$x' = x + p \quad (1.22)$$

$$y' = y + q \quad (1.23)$$

- **Rotación:** consiste en rotar la imagen cierto ángulo θ ver figura 1.4 y está definida por 1.24 y 1.25 :

$$x' = x \cos(\theta) + y \sin(\theta) \quad (1.24)$$

$$y' = -x \sin(\theta) + y \cos(\theta) \quad (1.25)$$

- **Rígida:** esta transformación incluye tanto rotación desde el centro de la imagen, como traslación y esta definida en la ecuación 1.26 ver figura 1.4.

$$T_u(x) = R(x - c) + t + c \quad (1.26)$$

Donde R es la matriz de rotación y t es de traslación y c es el centro de la rotación, este mismo modelo puede ser aplicado a un volumen de datos 3D, sin embargo hay que tener en cuenta la traslación en el eje Z y los ángulos de Euler o navegación 3D: cabeceo, alabeo, guiñada como se ve en la imagen 1.3.

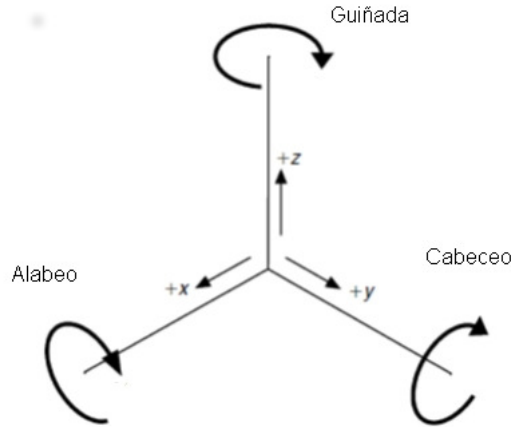


Figura 1.3: Ángulos de Euler o de navegación

Las ecuaciones para transformaciones 3D son para (x, y, z) :

$$\begin{aligned} en - x : x' &= x + p \\ en - y : y' &= y + q \\ en - z : y' &= z + r \end{aligned} \quad (1.27)$$

Para el cabeceo:

$$\begin{aligned} x' &= x \\ y' &= y \times \cos \theta + z \times \sin \theta \\ z' &= -y \times \sin \theta + z \times \cos \theta \end{aligned} \quad (1.28)$$

Para el alabeo:

$$\begin{aligned} x' &= x \times \cos \omega - z \times \sin \omega \\ y' &= y \\ z' &= x \times \sin \omega + z \times \cos \omega \end{aligned} \quad (1.29)$$

Y para el guiñada:

$$\begin{aligned} x' &= x \times \cos \phi + y \times \sin \phi \\ y' &= -x \times \sin \phi + y \times \cos \phi \\ z' &= z. \end{aligned} \quad (1.30)$$

Sin embargo es mejor utilizar una notación matricial con lo cual la transformación espacial queda expresada en términos de los parámetros p, q, θ si es el caso de 2D:

$$\text{en } x : \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & p \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1.31)$$

$$\text{en } y : \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & q \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1.32)$$

$$\text{Rotacion} : \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1.33)$$

Al multiplicar las matrices 1.31,1.32,1.33 se obtiene una matriz que rota y traslada en x y y . Se debe tener en cuenta que el orden en que se multiplica afecta el resultado de la matriz resultante, pero en el proceso de transformación espacial es igual no importa el orden ver [16]. La notación matricial puede aplicarse a cualquier transformación

- **Similitud:** la imagen puede trasladarse, rotar y cambiar de escala, está definida en la ecuación 1.34, ver figura 1.4.

$$T_{\mu}(x) = sR(x - c) + t + c \quad (1.34)$$

Con s un escalar que indica la escala isotrópica de la imagen, R la matriz de rotación, t traslación y c el centroide de la imagen.

- **Afin:** está definida en la ecuación 1.35, ver figura 1.4.

$$T_{\mu}(x) = A(x - c) + t + c \quad (1.35)$$

Donde la matriz A no tiene restricciones. La imagen puede ser trasladada, rotada, escalada e inclinada.

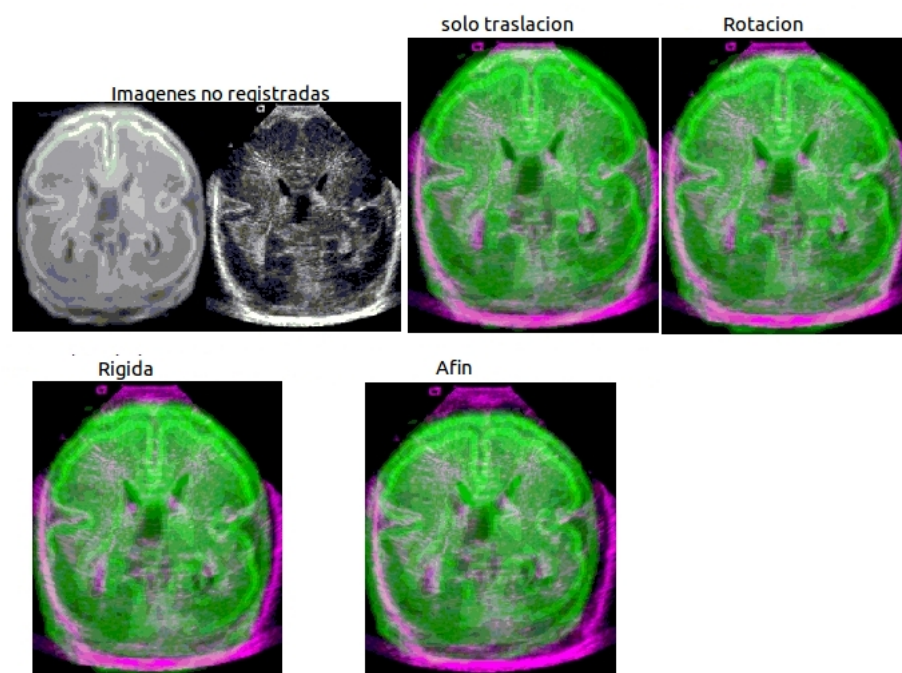


Figura 1.4: : Imágenes de un cerebro prematuro usando Resonancia y ultrasonido de un niño cortesía de [18] las dos imágenes han sido fusionadas y dado un color específico para apreciar el registro

- **B-spline:** esta transformación es del tipo no rígida y puede deformar la imagen objetivo para adecuarse a la imagen de referencia esta definida en la ecuación 1.36:

$$T_{\mu}(x) = x + \sum_{x_k} p_k \beta^3 \left(\frac{x - x_k}{\sigma} \right) \quad (1.36)$$

Donde X_k son los puntos de control, β^3 es un polinomio B-spline multidimensional cúbico, p_k son los coeficientes de la B-spline. El objetivo de esta transformación es deformar o mover la malla formada por los puntos de control en la imagen objetivo.

- **Thin plate spline:** es otra transformación no rígida. Usa un conjunto de K marcas en la imagen objetivo: X_k^{fix} y x_k^{mov} con $k = 1, \dots, K$. La transformación es expresada como la suma de una transformación afín con un componente no rígido como se ve en 1.37.

$$T_{\mu}(x) = x + Ax + t + \sum_{x_k^{fix}} c_k G(x - x_k^{fix}) \quad (1.37)$$

Donde $G(r)$ es la función básica y c_k son los coeficientes correspondientes a cada marca y estas se mueven por el vector paramétrico μ

Interpoladores

En las operaciones de transformación es en ocasiones necesario usar alguna técnica de interpolación. Este paso es un necesario cuando los puntos en una imagen son mapeados a posiciones no rígidas después de la transformación, la interpolación es realizada para aproximar los valores de los puntos transformados, ver [7]. Entre las más comunes tenemos:

- **Vecinos más cercanos:** utiliza el valor de los vecinos más cercanos al voxel o pixel para obtener así obtener un valor aproximado del punto interpolado.
- **Interpolación bilineal:** usa una interpolación lineal pero aplicada a dos variables, utiliza 4 vecinos más cercanos para sacar una media del valor. La interpolación Lineal utiliza dos puntos conocidos para obtener un tercero interpolado de los dos.
- **Convolución cubica:** similar a la bilineal pero utiliza 16 vecinos.

1.1.3. Optimizadores

Cada algoritmo de registro requiere un algoritmo de optimización, el cual busca la óptima transformación espacial para minimizar o maximizar una función de coste o medidas de similitud, ver [7]. Un algoritmo de optimización trata buscar cuál es el valor donde es mínimo cierta función ya sea en una o en varias dimensiones o con ciertas condiciones (restricciones). Para un algoritmo de optimización dado se puede expresar como 1.38

$$T_{optimo} = \arg \min_{(T)} f(T(X_s), X_R) \quad (1.38)$$

Donde T es la transformación y f es la función de coste a ser minimizada o maximizada. Existen varios métodos y se pueden dividir en optimizadores globales o locales. Los locales solo pueden encontrar valores mínimos localizados dado ciertas valores iniciales, la búsqueda no siempre hallará el valor mínimo de toda la función. Los optimizadores globales tratarán de buscar siempre el mínimo global en toda la función.

Algunos métodos son los siguientes:

- **Métodos basados en gradientes:** son generalmente usados para buscar la dirección en la cual el valor de la función de costo decrecerá localmente. El vector gradiente de la función de costo $f(x)$, donde \vec{x} es un vector n -dimensional $\vec{x} = [x_1, x_2, \dots, x_n]^T$ y puede ser expresado como 1.39

$$\nabla f(x) \equiv g(x) \equiv \left\{ \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right\}^T \quad (1.39)$$

Y en derivadas parciales de segundo orden puede expresarse por la matriz hessiana 1.40

$$\nabla^2 f(x) \equiv H(x) \equiv \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}^T \quad (1.40)$$

Entre los métodos basados en gradientes más usados tenemos al gradiente descendiente: Donde usa la expansión de Taylor de primer orden para aproximar la función y como resultado el vector gradiente es usado como la dirección de búsqueda así: 1.41.

$$\vec{p}_k = -\nabla f(\vec{p}_k) \quad (1.41)$$

- **Métodos de Newton y cuasi-Newton:** el método de Newton, una expansión de segundo orden es usada para aproximar la función, y buscar la dirección donde puede ser obtenida la solución a la ecuación 1.42 :

$$H(\vec{x}_k)\vec{p}_k = g(\vec{x}_k) \quad (1.42)$$

Los métodos cuasi-Newton aproximan y actualizan la matriz hessiana en cada iteración. Existen dos implementaciones muy eficientes: algoritmo de Dividon-Fletcher-Powell donde la inversa de la matriz hessiana es calculada, y la otro es el método de Broyden-Fletcher-Goldfarb-Shanno donde una aproximación a la matriz hessiana es usada.

Algunos algoritmos de optimización están basados en comportamientos biológicos, como por ejemplo el proceso de alimentación o de cría de ciertas especies de animales. Estos han inspirado métodos de búsqueda para la solución de problemas de optimización como se ve en [19]. Estos algoritmos reciben el nombre de algoritmos evolutivos. Algunos de ellos son:

- **Evolución diferencial:** es un tipo de algoritmo evolutivo para búsquedas sobre espacios continuos desarrollado por Storn y Price como se ve en [19]. En este método una población inicial es generada aleatoriamente. En cada iteración se generan una nueva población evolucionada de la anterior usando un vector diferencial calculado por el número de otros individuos. Si la nueva población es mejor que la anterior, la nueva población la reemplazará, si no es descartada.

- **Algoritmos Genéticos:** es uno de los tipos de algoritmos evolutivos mas viejos y comunes desarrollado en los años 70 como se ve en [19]. En su forma mas simple, una población inicial es seleccionada aleatoriamente en el espacio de búsqueda y usando sus mutaciones y cruces para generar la siguiente población. La modificación de las mutaciones y cruces hacen que la siguiente población sea mejor o peor. La más común de los cruces es simplemente tomar dos parámetros y promediarlos. La mutación generalmente consiste en tomar un parámetro y reemplazarlo con una perturbación.
- **Búsqueda Cuco (Cuckoo search (CS)) :** este método de optimización metaheurístico esta basado en el comportamiento de algunas especies de pájaros conocidas como cucúlidos entre los cuales se encuentran los cucos. Desarrollado por Xin She Yang y Suash Deb en [20]. Los pájaros cucos tiene una estrategia de reproducción muy agresiva, varias especies tales como el ani y el Guira ponen sus huevos con nidos comunitarios, donde ellos pueden remover otros huevos para incrementar la probabilidad de eclosionar de sus propios huevos. Otras especies ponen sus huevos en otros nidos de una especie anfitrión los cuales se convierten en parásitos de la especie anfitrión. Algunas de estas especies anfitrionas entran en conflicto con los cucos invasores y en algunos casos llegan a remover el huevo parásito o abandonan su nido y lo hacen en otra parte. Sin embargo alguna especie de cucúlidos imitan en patrón y colores a los huevos de la especie anfitrión.

Búsqueda cuco (CS) utiliza los siguientes representaciones: Cada huevo en un nido representa una solución. El objetivo es utilizar las nuevas y potencialmente mejores soluciones (cucos) para reemplazar una no tan buena solución en los nidos, en la forma más simple cada nido tiene un huevo. El algoritmo se puede extender a casos más complicados en los que cada nido tiene múltiples huevos que representan un conjunto de soluciones.

- **Métodos de inteligencia de enjambre:** estos algoritmos surgen del estudio del comportamiento de insectos sociales tales como las abejas o hormigas o luciérnagas y usan el modelo ese comportamiento para resolver problemas de optimización como se aprecia en [19]. Sistemas basados en inteligencia de enjambre, son hechos de unas poblaciones de simples individuos que interactúan con otros en sus alrededores siguiendo unas reglas simples de comportamiento. Cada individuo o partícula son ignorantes de su propia naturaleza, el comportamiento colectivo tiende a una inteligencia global colectiva. Entre algunos de ellos tenemos:

- **Algoritmo de optimización por enjambre de partículas** Basic Particle Swarm optimization (PSO) : es un método de optimización heurístico propuesto por Eberhart y Kennedy como se aprecia en [21]. Está inspirado en el comportamiento social de agentes encontrados en la naturaleza. Este comportamiento puede observarse en pájaros sociales , enjambres de abejas, escuelas de peces entre otros. Por ejemplo un enjambre de abejas en busca de alimento, cada abeja busca aleatoriamente por el espacio tratando de localizar la mejor región con la mayor densidad de flores, ya que se supone que ahí hay mas polen. Cada abeja transmite sus hallazgos a la demás abejas y así el enjambre sabrá colectivamente cuál es la mejor región cambiando la búsqueda de las demás abejas para concentrarse en ese punto. La búsqueda puede refinarse o cambiar si se encuentra un mejor espacio.

El algoritmo PSO es una técnica de inteligencia de enjambre basado en computación evolutiva, como se explica en [22] cada individuo en el enjambre es una partícula de menos volumen en un espacio de búsqueda multidimensional. La posición en el espacio de búsqueda representará una posible solución al problema de optimización, y la velocidad de vuelo determina la dirección y paso de la búsqueda. La partícula vuela en el espacio de búsqueda a una determinada velocidad que es ajustada dinámicamente acorde con su propia experiencia y de las otras partículas. Constantemente ajustan sus dirección y

velocidad en busca de la mejor posición encontrada. Gradualmente se moverán a una región mejor y finalmente llegaran a la mejor posición en todo el espacio de búsqueda.

- **Algoritmo Firefly (luciérnagas):** las luciérnagas son una especie de insecto que es capaz de producir una luz natural para atraer a su presa o a su pareja. Existen alrededor de 2000 especies de luciérnagas que producen cortos y rítmicos destellos que poseen un único patrón por especie como se ve en [23].

La intensidad (I) de los destellos decrece con el incremento en la distancia (r) esto hace que solo se puedan comunicar por algunos cientos de metros. La formulación de los algoritmos de optimización está basado en 3 ideas generales:

1. Una luciérnaga se sentirá atraído por otras luciérnagas, independientemente de su sexo.
2. La atracción es proporcional a su brillo y decrece con la distancia a medida que ella crece.
3. La forma de la función objetivo determina el brillo de una luciérnaga.

El algoritmo Firefly esta basado en la fórmula física de la intensidad de la luz I que decrece con el incremento del cuadrado de la distancia r^2 . Sin embargo como la distancia desde el origen de la luz se incrementa, la absorción de la luz hace que la luz se vuelva cada vez mas débil. Como sucede en la analogía con las luciérnagas serán atraídas por la luz y la intensidad de esta representa una solución del problema de optimización, el algoritmo buscará la mejor población en el espacio de búsqueda. El algoritmo puede ser mejorado usando en vez de un paso aleatorio una técnica conocida como el vuelo de Lévy, la cual forma unos pasos aleatorios pero siguiendo una distribución de probabilidad Heavy-tailed (cola pesada) como se ve en [24].

Entre otros tipos de algoritmos de optimización tenemos:

- Búsqueda de patrones (Pattern search) de Hooke-Jeeves.
- Recocido simulado (Simulated Annealing).
- Método de Nelder Mead.

Por lo tanto los pasos necesarios para un registro de imágenes se puede resumir en la imagen figura 1.5.

1.1.4. Fusión de imágenes

Es el proceso de combinar la información de dos o más imágenes en una misma escena para mejorar la visión y el entendimiento de la esta ver imagen 1.4. Generalmente se fusiona imágenes registradas pues se puede hacer pixel a pixel. $I_1(x, y), I_2(x, y), \dots, I_t(x, y)$ representan T imágenes de tamaño $M \times N$ capturando la misma escena, cada imagen es adquirida usando diferentes instrumentos o técnicas, en consecuencia cada imagen tiene sus propias características visuales según [25].

Las técnicas de fusión generalmente se dividen en técnicas de dominio espacial y de transformación. En las técnicas de dominio espacial, las imágenes son fusionadas usando característica

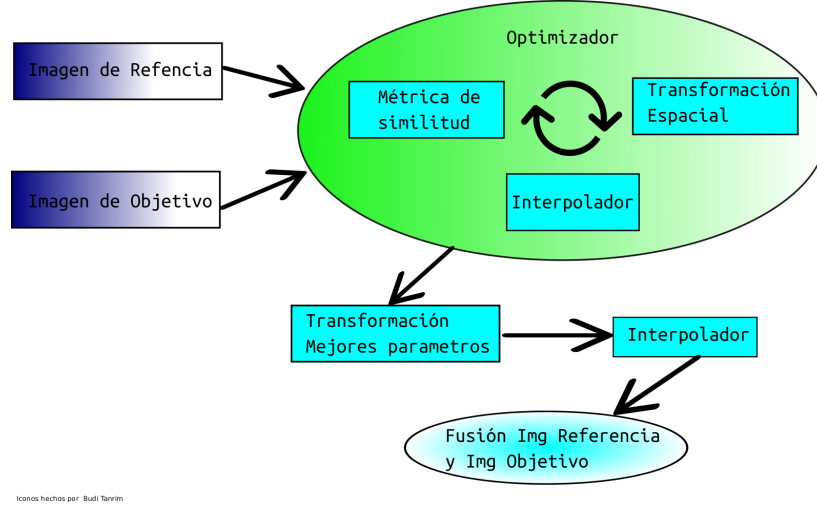


Figura 1.5: se realiza una métrica de similitud a la imagen objetivo y de referencia, se realiza una optimización y así encontrar el mínimo de la función métrica, se realiza la transformación y una etapa de interpolación, el proceso se itera hasta encontrar la transformación correcta que alinee las imágenes

espaciales. Asumiendo que $g(\cdot)$ representa alguna regla de fusión, la técnica que combina las características de las imágenes puede expresar:

$$I_f(x, y) = g(I_1(x, y), \dots, I_t(x, y)) \quad (1.43)$$

Se puede aplicar una transformación a las imágenes para descubrir las estructuras básicas de estas y así la fusión se expresa en la ecuación 1.44. Donde T representa algún operador de transformación y $g(\cdot)$ la regla de fusión aplicada.

$$I_f(x, y) = \tau^{-1} [g \{ \tau(I_1(x, y)), \dots, \tau(I_T(x, y)) \}] \quad (1.44)$$

Como las imágenes están registradas, la fusión se puede hacer pixel a pixel, entonces algunas reglas de fusión según [25] son:

- **Fusión por promediación:** se obtiene promediando los correspondientes coeficientes en cada imagen.
- **Fusión por el máximo absoluto:** fusiona por selección los mas grandes en valor absoluto de los correspondientes coeficientes en las imágenes.
- **Fusión por eliminación de ruido:** realiza simultáneamente una fusión y eliminación de ruido por la umbralización de los coeficientes de transformación.
- **Fusión alto/bajo (High/low):** combina las partes de alta frecuencia de algunas imágenes con las partes de baja frecuencia de otras.

1.1.5. Técnicas para la evaluación del registro

Desde el punto de vista del usuario la precisión o exactitud es más importante en si que el método de registro obtenido, sobretodo en medicina donde el conocimiento absoluto de la precisión es necesario para tomar decisiones apropiadas. Cinco criterios se pueden evaluar para registro de imágenes médicas según [9]

- **Precisión:** se refiere a la diferencia en distancias entre la imagen objetivo y la de referencia. Generalmente se escogen imágenes en las cuales las distancias son conocidas luego se saca el error en la distancia entre puntos conocidos, ese error puede ser medio, máximo o medio cuadrático RMS.
- **Confiabilidad:** trata del número de veces que el algoritmo encuentra una respuesta satisfactoria comparada con el número total de pruebas realizadas. Si se hacen n pruebas a n pares de imágenes y m de n son el número de veces que el algoritmo encuentra una respuesta satisfactoria, si n es lo suficientemente grande entonces m/n representara la confiabilidad del algoritmo. Entre más cerca este el radio a 1 más confiable será el algoritmo.
- **Robustez:** es el grado de estabilidad del algoritmo bajo condiciones cambiantes de los parámetros de entrada. Puede ser evaluado con respecto al ruido, intensidad, diferencias geométricas, oclusión, ect. Por lo general si un algoritmo es poco preciso y no confiable también se dice que es poco robusto.
- **Complejidad computacional:** la cual determina la rapidez del algoritmo y muestra cuán práctico es el algoritmo en situaciones reales. La complejidad computacional es la medida del número de adiciones/multiplicaciones expresadas como función de las dimensiones de la imagen. Para un algoritmo de registro es deseable que tenga una complejidad computacional que sea función lineal de las dimensiones de la imagen.

Evaluar estos parámetros dentro de un método de registro en lo general no es una tarea fácil según [26], pues no existe un estándar general con lo cual se pueda realizar con lo que muchas técnicas de validación se a reportado en la literatura como se aprecia en el estado arte, sin embargo es difícil establecer comparaciones entre ellos pues resultan incompatibles metodológicamente.

Unidades para el reporte del error en el registro.

Las unidades que se adoptan para informar el registro van a depender del contexto en el cual el método fue usado pero generalmente expresan la diferencia promedio entre las distancias de la locación del objeto o estructura expresada ya sea en milímetros, centímetros, metros, pixeles, ect. El error medio cuadrático (RMS) también se usa como alternativa al error medio. Una gráfica de la distribución de los errores en el punto de interés puede ser de más ayuda y un test estadístico como el de Kolmogorov-Smirnov puede ser usado para comparar las distribuciones de los errores como se aprecia en [26].

Para registros rígidos los errores generalmente se informan dada la transformación espacial (traslación en x , rotación, ect), esto se puede comparar entre métodos pero se debe tener cuidado pues al aplicar una matriz de transformación la descomposición en los elementos básicos como rotación y traslación no siempre es única y por lo tanto errores reportados por un autor pueden no ser comparables con los de otro autor. Tal vez la forma más trivial de buscar desalineamientos en imágenes que se registran es usando inspección visual, pero como se ve en [26] se puede detectar

desalineamientos de 2 mm y 2 a 4 grados en imágenes de MRI o CT. Por lo general si una imagen parece desalineada de seguro no está bien registrada. Por último se acostumbra para probar la valides de los algoritmos de registro usar simulaciones, Phantom o cadáveres.

1.2. Imágenes diagnósticas tipo DICOM tipos usados y características

1.2.1. Técnicas de imágenes diagnósticas

Se considero explorar el uso de cuatro tipos de imágenes diagnosticas tipo DICOM para realizar el registro entre las cuales tenemos:

- Imágenes por Resonancia magnética (IRM)
- Tomografía axial computarizada (TAC o simplemente CT)
- Fluoroscopia y angiografía
- Ultrasonido

El uso de estos tipos en específico de imágenes esta determinado por las necesidades propias del sistema que se esta usando, mas no por algún tipo de restricción a solamente esos tipos de imágenes, las técnicas de registro usadas son independientes del tipo de imágenes, solo cambia cierta manipulación inicial de la imagen.

1.2.2. Tipos de imágenes usadas

Imágenes por Resonancia magnética (IRM): Es una técnica para la obtención de imágenes médicas que utiliza la resonancia magnética para obtener información de las estructuras corporales. La resonancia magnética está basada en propiedades mecánica-cuánticas de los átomos. Núcleos atómicos que contengan un número impar de nucleones (protones y neutrones) tienen un momento magnético como algunos isotopos del Hidrógeno (protio H1 y tritio H3) tienen un gran momento magnético. En ausencia de un campo magnético externo, el momento magnético de estos núcleos apuntaría en varias direcciones, pero en la presencia de este los momentos magnéticos tienen a alinearse con él, por lo tanto se puede obtener la energía del núcleo con un momento magnético m en presencia de un campo magnético B . Luego se aplica un pulso de radiación electromagnética a una determinada frecuencia de resonancia el cual afecte a los núcleos atómicos estos reaccionan cambiando su estado energético pero retornan a su estado inicial liberando fotones los cuales puedes ser captados y con esto crear una imagen a partir de la densidad de protones en una sección transversal del paciente [27].

Tomografía axial computarizada (TAC o simplemente CT): Tomografía viene del griego Tomos que significa sección, corte o rodaja y por lo tanto es el proceso de formación de imágenes creada a partir de secciones transversales. La tomografía computarizada usa rayos X para crear estos corte perpendicular a los pacientes. Numerosos cortes son tomados en alguna dirección longitudinal,

luego estos cortes se ensamblan y se puede obtener una información tridimensional de dichos cortes. Los rayos X es uno de los más viejas fuentes de radiación electromagnética usada para imágenes diagnósticas. Son producidos cuando en un tubo al vacío con un cátodo y un ánodo, el ánodo es calentado causando que los algunos electrones se liberen. Este flujo de electrones viaja a gran velocidad hacia el cátodo positivo. Cuando los electrones golpean el núcleo atómico, la energía es liberada en forma de radiación X la cual posee una longitud de onda entre 0.01 a 10 nanómetros y son altamente penetrantes y pueden traspasar fácilmente estructuras corporales blandas como piel o tejido [27].

Fluoroscopia y angiografía : Sistemas de fluoroscopia producen proyecciones de imágenes en tiempo real de rayos X con una alta resolución y pueden adquirir continuamente. Estos sistemas son usados para tener un sistema de imagen que puede guiar al especialista en intervenciones en los pacientes como es el caso de la angiografía. Sistemas de fluoroscopia moderna usan intensificadores de imágenes acoplados a un sistema de video digital o algún detector plano. Su principal diferencia entre el fluoroscopia y los sistemas de rayos X convencionales es que pueden adquirir imágenes en tiempo real y dosis de radiación mucho mas bajas [28].

La angiografía es una técnica para el estudio de venas y arterias usando un fluoroscopia. Para esto se utiliza algún medio de contraste que es ingresado al paciente y por lo tanto los rayos X no pueden atravesar este medio de contrastes radiopaco y los vasos sanguíneos salen en dicha radiografía. [28]

Formación de imágenes diagnósticas usando ultrasonido: Ultrasonido es una vibración mecánica con una frecuencia arriba de rango de audición humana. La formación de imágenes médicas por ultrasonido se basa en el hecho de enviar ondas sonoras de alta frecuencia que hacen eco en las estructuras corporales y un computador recibe dichas ondas reflejadas y las utiliza para crear una imagen. Las ondas producidas pueden ser directamente acopladas a los tejidos donde se propagan a la velocidad del sonido, en su viaje pueden sufrir varios fenómenos como absorción, refracción , reflexión y dispersión. Sin embargo estas ondas se reflejan y son captadas por el transductor. Imágenes por ultrasonido es uno de los más usados procedimientos para obtener imágenes médicas. La principal ventaja sobre otros métodos es que es poco invasivo y mucho más seguro pues no utiliza radiación ionizante [29].

Los dispositivos de ultrasonido médico utilizan ondas ultrasónicas a una frecuencia de 1-15Mhz. La transmisión de las ondas generalmente se hace en forma de un tren de pulsos cortos, propagándose en el tejido donde se refleja en las estructuras y retorna a la fuente de los ecos. Esta señal es inicialmente amplificada y filtrada análogamente y luego digitalizada usando un CAD (Convertor análogo digital) generalmente con una resolución de 8-14 bits. La señal de alta frecuencia recibida el eco es amplificada y la señal es modulada en fase a una señal portadora. La señal es de modulada para obtener la frecuencia base.

El neuronavegador, ver [30], es un sistema basado en la localización intraoperatoria guiada por imagen, sin marco estereotáxico. Consiste en un método de registro de la imagen y espacio físico, un instrumento de localización, y una visualización de las imágenes médicas obtenidas y procesadas por un ordenador. Este sistema proporciona una retroalimentación y visualización, en tiempo real, de la localización de los instrumentos quirúrgicos. Un puntero con un sistema de marcadores pasivos, permite guiar sobre una imagen espacial virtual reconstruida a partir de las imágenes de TAC o RMN preoperatorias del paciente.

La navegación se basa en dos cámaras ubicadas con posiciones diferentes con los cuales se puede obtener información 3D por medio de ciertos algoritmos de visión estéreo que son capaces

de reconocer el espacio físico y así dar una idea de la localización de los instrumentos como se ve en la figura 1.6. Aunque como tal no es un sistema de adquisición de imágenes diagnosticas pero sirve para relacionar esas imágenes con el mundo real.



Figura 1.6: Neuronavegador creado por el grupo Applied Neuroscience

1.2.3. El estándar DICOM

Desde el momento en que se produce el intercambio de datos digitales en el sector salud, nace la necesidad de estandarizar el almacenamiento, la transmisión y la manipulación en general de los mismos para garantizar que todos los dispositivos de visualización desplieguen la misma información a los especialistas de manera fidedigna. DICOM (Digital Imaging and Communications in Medicine ISO 12052) es un estándar ver [31] creado y administrado por la National Electrical Manufacturers Association - NEMA <http://dicom.nema.org/>, que permanece en constante actualización, adecuándose a los nuevos avances en el sector Salud. El propósito principal del estándar es garantizar un entorno controlado de condiciones propuestas para las imágenes médicas, cubriendo desde el momento en que se adquiere un estudio imagenológico hasta el momento de ser desplegado en pantalla o impreso en papel radiográfico como se ve en [30].

NEMA presenta un conjunto de 18 partes que se actualizan periódicamente como se ve en [31] y que cubren la siguiente temática:

1. Introducción y recapitulación.
2. Conformidades.
3. Definiciones de objetos de información.
4. Especificaciones de clases de servicio.
5. Codificación y estructura de datos.
6. Data Dictionary.
7. Intercambio de mensajes.
8. Redes de comunicación de soporte para el intercambio de mensajes.

9. Medios de almacenamiento y Formato de archivos para el intercambio de datos.
10. Perfiles de aplicaciones para el almacenamiento de archivos.
11. Formatos de medios y medios físicos para el intercambio de datos.
12. Función de escala de grises estandarizado para despliegue en pantalla.
13. Perfiles de seguridad.
14. Contenido de recursos de asignación.
15. Información explicativa.
16. Acceso Web a objetos DICOM persistentes.
17. Aplicación Hosting.
18. Transformación de DICOM a y desde Estándares HL7.

Se convierte en un requisito la inclusión de las componentes principales del estándar para la extracción de la imagen de los archivos entregados por los dispositivos imagenológicos. Para una aplicación designada para la visualización de las imágenes, se debe considerar el tipo de variable física representado y las características espaciales de la escena para poder determinar la decodificación de los datos almacenados; según el tipo de tareas a realizar es posible obviar la inclusión de PARTES del estándar.

1.2.4. Estructura de un archivo DICOM

Para introducirse en el complejo entorno de la medicina, DICOM usa su propio lenguaje basado en un modelo propio del mundo real. Se define el mundo real como todos los datos físicos o descriptivos; por ejemplo el nombre del paciente, el dispositivo médico usado, la orientación del paciente, los parámetros de adquisición de la secuencia, la imagen digitalizada, etc. que son vistos por DICOM como elementos con sus respectivos atributos y propiedades [30]. Es así como se establece una jerarquía entre los datos a través de una clasificación por grupos según el contenido de los elementos, facilitando la identificación y el acceso a las variables dentro de un mismo archivo.

Por lo general, un archivo DICOM se reconoce por su extensión *.dcm; sin embargo esto no es una exigencia del Estándar, por lo que la forma de diferenciarlo es por medio del HEADER o cabecera, que consta de 128 bytes a modo de preámbulo y 4 bytes con el prefijo "DICM". El preámbulo puede estar en blanco o contener información sobre la aplicación principal con la que debe ser abierto el archivo.

El cuerpo del archivo se forma por una secuencia de Data Sets, que representan objetos del mundo real y que a su vez están constituidos por Data Elements, que son atributos codificados del objeto (Figura 2.10). Cada Data Element es identificado y clasificado por un TAG o etiqueta. Una etiqueta es un identificador único para un Data Element y se compone de dos partes. DICOM utiliza la siguiente notación en hexadecimal para referirse a una etiqueta: (gggg,eeee) el primer valor 2 bytes es el número de grupo y el segundo es el número del elemento. Por ejemplo (0010,0030) corresponde a la fecha de nacimiento del paciente.

Además del TAG, el Data Element está compuesto por otras tres propiedades: El “Valor de Representación” (VR) indica el tipo de dato que se tiene almacenado; la “Longitud” especifica el tamaño en bytes ocupado por el Data Element, y finalmente se tiene el “Valor” o el dato almacenado (Value Field) como se ve en [31] en la parte 6 (Data Dictionary).

Existe una gran variedad de Data Elements y se debe tener en cuenta que no siempre estarán definidos en su totalidad dentro de un mismo archivo, la presencia de estos dependen del tipo de estudio. Asimismo, habrá Data Elements que no aporten información relevante para ciertas necesidades aunque se encuentren presentes.

1.2.5. Comunicación entre dispositivos DICOM

En el estándar DICOM [31] en la parte 7 (Message Exchange) y 8(Network Communication Support for Message Exchange) se define como debe ser la comunicación para la transmisión de archivos tipo DICOM entre dispositivos que cumplen con dicho estándar. Para ese fin el Nema a creado ciertos protocolos de comunicación de red basados en TCP/IP. En la parte 7 del estándar DICOM se encuentra el DICOM Message Service Element (DIMSE) donde se especifica los servicios y protocolos usados por una aplicación de imágenes medicas para intercambiar mensajes ha través de los servicios de comunicación soportados por el estándar. Un mensaje esta compuesto por una secuencia de comandos seguidos por una secuencia de datos .Los conceptos utilizados en el DIMSE fueron desarrollados a partir de estándares internacionales como el ISO 7498-1 (Modelo OSI) y el ISO/TR8509 como se aprecia en [32].

El modelo de referencia OSI es usado para modelar la interconexión de los equipos de imágenes medicas, el cual cuenta con siete capas de comunicación. DICOM usa la capa OSI Upper Layer Service para separar el intercambio de mensajes DICOM en la capa de aplicación del soporte de comunicación proveída por las capas inferiores.

El protocolo para la comunicación DICOM es muy similar al protocolo TCP/IP, a cada dispositivo DICOM se le asigna una dirección IPv4 pero además de eso se le asigna un nombre específico llamado DICOM Application Entity (abreviado AE o AETitle) para distinguirlo de otros dispositivos o servicios , pues estos pueden compartir una misma IP pero prestar diferentes servicios, ejm un DICOM server, una estación de guardado, una impresora o un dispositivo de adquisición. También se configura un puerto de comunicación generalmente 104 o 11112 (puerto estándar) pero puede ser cualquiera entre 0-65535 evitando puertos conocidos como el 80 (HTTP), 23 (Telnet), 21 (FTP). Un AETitle es una identificación para una aplicación no para un computador específico, pues en un computador pueden coger diferentes servicios DICOM.

El protocolo de comunicación que ofrece DICOM opera de la siguiente manera: un AE solicita un servicio (Service Class) y es llamado Service Class User (SCU) y es procesado por un AE prestador de servicios conocido por Service Class Provider (SCP).Otra definición importante son los Service Object Pair (SOP), que son la unión de un Information Object Definition (IOD) y los DIMSE. La clase SOP contiene las reglas y semánticas para usar los DIMSE. Ejm de DIMSE son guardar, mover, buscar, ect. Ejms de Objetos son imágenes CT, MR, US, entre otras o lista de impresión o tareas, ect.

Entre los servicios (DIMSE) mas comunes que puede solicitar un SCU o proveer un SCP se encuentran [32]:

- C-ECHO (Verification SOP): es el servicio DIMSE mas simple y común, es usado para comprobar la conexión entre AE diferentes. Un AE solicita un C-ECHO (C-ECHO-Rq del ingles request petición), si un AE responde un con mensaje válido (C-ECHO-Rsp del ingles reply responder), los dos AE están debidamente conectados. En algunos dispositivos también esta la opción de mandar un ping para comprobar conexiones.
- C-FIND (FIND SOP): Es el encargado de buscar datos DICOM, en un específico AE. Una de las principales ventajas sobre un simple protocolo de búsqueda es la habilidad de encontrar por un dato DICOM específico, usando los mensajes propios de C-FIND. Aunque no usa un específico SOP para cada modalidad como C-Store, DICOM divide todos las búsquedas en 3 niveles de datos: Paciente, Estudio y Paciente-Estudio, cada una de ellas tiene especificas Tags que se pueden utilizar como parámetros de búsqueda estos niveles se llaman Roots y C-find tiene un separado SOP para cada nivel: ejm si escogemos el nivel paciente (Patient Roots model), podemos hacer una búsqueda por el tag (10,10) (nombre del paciente) y se puede refinar la búsqueda por el tag (10,20) (ID del paciente), si vamos por el nivel estudio se puede buscar por el tag (20,10) (ID del estudio) e incluir también el tag (10,10) pues también esta definido en el nivel, sin embargo el tag (10,20) no estará disponible para la búsqueda en ese nivel. C-Find admite en su búsqueda caracteres comodines como *, por ejm si se desea buscar todos los nombres de paciente que empiezan con F se colocaría F*.
- C-STORE (Store SOP): Es el encargado de mover las imágenes DICOM y otros tipos de datos entre los AEs a través de la red DICOM. Debido a que diferentes objetos implican diferentes procesos, DICOM asigna una separada SOP a cada clase de modalidad por ejemplo almacenamiento de imágenes CT (CT image Store SOP), o almacenamiento de imágenes de MR (MR image store SOP). Por eso DICOM Store es representado por una familia de específicos storage SOPs cada uno con su único identificador (unique identifier UID).
- C-GET (GET SOP): Básicamente consiste en mover entre Aes archivos DICOM, conceptualmente C-GET tiene incluido un C-Find y un C-Store en una sola clase de servicio. Los parámetros de búsqueda son similares a los utilizados por C-Find, pero no se admiten caracteres comodines.
- C-MOVE (MOVE SOP): Es el servicio DIMSE más usado para mover archivos entre AE, prácticamente es el mismo C-Get, pero un poco más complejo, seguro y de mas utilidad. La diferencia consiste en que C-GET va a mover el archivo al AE solicitante en cambio C-MOVE puede mover a otro diferente AE, además C-MOVE necesita saber quien le está mandando la solicitud para poder activarla, esto genera que C-MOVE sea más seguro que C-GET, el cual moverá el archivo hacia el solicitante cualquiera que fuese este. Por lo tanto C-GET a sido declarado por Mena como obsoleto y ya no tiene mucho soporte en algunos servers.

Servidor PASC (Picture Archive and Communication Systems)

Para correcta implementación de la comunicación entre dispositivos DICOM es necesario el uno de los PASC los cuales son un sistema de información muy útil para todos los elementos dentro de esa comunicación. En si los PASC son componentes claves en redes clínicas pues proveen el hardware y el software necesarios [33] para todas las transacciones de imágenes en entornos médicos tales como adquisición, archivado, distribución y visualización a través de redes de computadores. Los servidores PACS normalmente cohabitan con otros medios de información tales como servidores HIS (hospital information systems) o RIS (radiological information system).

Los PACS fueron originalmente diseñados para interpretar imágenes mas fácilmente en departamentos de radiología, sin embargo a evolucionado ha ser una herramienta integral hospitalaria que genera un sistema de información mucho mas allá de lo referente a radiología como se ve en [34]. Y en la ultima década a soportado la expansión a nuevas herramientas para asistir a las imágenes diagnósticas, como teleradiología y diagnósticos asistidos por computadora CAD (Computer-Assisted Diagnosis)

Un PACS puede ser organizado en: productores de imágenes, consumidores de imágenes, infraestructuras de redes, servidores operacionales y de archivado. Según el servidor Dicom <http://www.pacsone.net/> La mayoría de los PACS consta de las siguientes modalidades según :

- Modalidades de entrada: Son los dispositivos de imágenes médicas como CT, MRI ,CR, Ultrasonido, ect y son donde las peticiones de C-STORE provienen.
- Servidor DICOM: Son programas o servicios que esperan por las modalidades de entrada, estaciones de trabajo o otros PACS para mandar imágenes DICOM.
- Servidores PACS : estos programas consisten en una base de datos que rastrea toda imagen en el PACS. También se encarga de recibir las imágenes.
- Servidor de archivos: este servidor se encarga de almacenar las imágenes a largo plaza en otros modalidades como otros discos duro, o DVD o CDs.
- Servidor Web : Es una manera económica de distribuir las imágenes a los médicos o especialistas
- Interfaces a los RIS: provee una interfaces entre la base de datos RIS y PACS usando comunicación HL7
- Clientes Radiológicos: son sistemas avanzados de computo para mostrar y manipular las imágenes de radiología.

1.3. Estado del arte

Como era de esperar el campo de investigación acerca de registro de imágenes es muy grande y está contando con mucho desarrollo en las últimas décadas debido a gran parte al aumento de la capacidad computacional. Ciertamente son muchos los campos que se benefician con tales métodos obviamente incluyendo la medicina.

A continuación se presentaran algunos artículos e investigaciones que tratan diversos aspectos en la medicina. Primero se mostrarán algunos artículos donde se analizan como tal las técnicas para registro basadas en intensidad para imágenes médicas en general como en:

Intensity based image registration by maximization of mutual information. Registro de imágenes basada en intensidad maximizando la información mutua [35].

Autores: R Suganya, Dr S.Rajaram y K.Priyadharsini

Se presenta una técnica de registro de imágenes médicas de la cabeza basada en intensidad que ha sido desarrollado maximizando la información mutua. El principal objetivo del artículo

es incrementar la precisión del registro así como su tiempo de ejecución pues ha adquirido gran importancia en el diagnóstico, tratamiento, estudios funcionales y terapias guiadas por imágenes sobre todo en imágenes multimodales como TAC y RM. Los experimentos muestran que la técnica propuesta en el artículo es robusta y eficiente con resultados precisos.

An automatic image registration scheme using Tsallis entropy. Un esquema de registro automático de imágenes usando entropía de Tsallis [14].

Autores: Mohanalin, Beenamol, Prem Kumar Kalra, Nirmal Kumar.

En el artículo se plantea un método de registro para imágenes médicas usando entropía de Tsallis para reemplazar la entropía de Shannon generalmente usada para utilizar medidas de comparación entre imágenes usando información mutua. Sus autores establecen que los algoritmos actuales basados en la entropía de Tsallis no son del todo automáticos aunque así se afirme. En este artículo el método es evaluado usando una transformación afine de imágenes de mamografía adquiridas clínicamente y comparando los resultados obtenidos con el método tradicional usando información mutua normalizada con la entropía de Shannon. Sus algoritmos han presentado resultados prometedores incrementando su precisión y reduciendo el número de evaluaciones y sus autores han encontrado que este algoritmo es efectivo para reemplazar la entropía de Shannon en registro de imágenes usando transformaciones afines. Ahora se mostraran algunos artículos sobre registro de imágenes para intervenciones médicas combinando imágenes preoperatorias con intraoperatorias:

Registration of CT and Intraoperative 3-D Ultrasound Images of the Spine Using Evolutionary and Gradient-Based Methods. Registro de imágenes de tomografía y ultrasonido 3D intraoperatorio de espina usando métodos evolutivos y basados en gradiente [36].

Autores: Susanne Winter, Bernhard Brendel, Ioannis Pechlivanis, Kirsten Schmieder, Christian Igel

Los autores presentan una comparación de 3 algoritmos basados en gradiente y uno evolutivo para el registro de imágenes de tomografía y ultrasonido. El sistema fue desarrollado para inserción de los tornillos pediculares durante la cirugía de espina. Con datos preoperatorios e intraoperatorios se ha determinado la precisión del registro con rangos realistas de desalineamientos iniciales. Se observaron diferencias significativas entre los métodos de optimización tratados pero el método con el algoritmo evolutivo ha demostrado la mejor rendimiento fallando solo 4 de 12000 registros.

Automatic ultrasound–MRI registration for neurosurgery using the 2D and 3D LC2 Metric. Registro automático de ultrasonido con IRM para neurocirugía usando métrica 2D y 3D LC2 [37]

Autores: Bernhard Fuerst, Wolfgang Wein, Markus Muller, Nassir Navab

En este artículo sus creadores muestran un algoritmo para el registro de imágenes de resonancia magnética (IRM) y ultrasonido (US) que permita neurocirugías guiadas por imágenes pues es conveniente el alineamiento de imágenes preoperatorias obtenidas por IRM con imágenes intraoperatorias de ultrasonido. El algoritmo propuesto usa una métrica de similitud basada correlación lineal de combinaciones lineales (LC2) para alinear ya sea cortes o volúmenes de US con imágenes MRI. Este enfoque permite un registro robusto y automático el cual ha sido evaluado exhaustivamente mostrando una buena precisión y exactitud.

3D navigation and monitoring for spinal milling operation based on registration between multiplanar fluoroscopy and CT images. Navegación 3D y monitoreo para operaciones de fresado en la espina basado en registro entre fluoroscopia multiplanar y imágenes CT [38].

Autores: Sheng Luan, Tianmiao Wang^b, Weishi Li^c, Zhongjun Li^{uc}, Liang Jiang^c, Lei Hu^b.

En el artículo sus creadores presentan un algoritmo de registro 3D/3D entre imágenes preoperatorias de tomografía e imágenes intraoperatorias multiplanares de fluoroscopia para operaciones de fresado en la espina. El método de navegación 3D es introducido para mejorar este tipo de operaciones ya que requieren gran habilidad y experiencia del cirujano y así mejorar la seguridad del paciente. El registro de estas imágenes se hace a través de una métrica de similitud basada en intensidad, ambas imágenes tienen un sistema de referencia común en el espacio quirúrgico. Una región crítica es definida para el monitoreo en tiempo real y así evitar que se afecte regiones nerviosas con diversas advertencias sonoras y visuales. Los autores demostraron que este sistema mejora la seguridad en este tipo de intervenciones.

An Image Registration Method for Head CTA and MRA Images Using Mutual Information on Volumes of Interest. Un método de registro de imágenes para angiografía por tomografía computarizada y angiografía por resonancia para la cabeza usando información mutua en volúmenes de interés [39]

Autores: Yutaro Yamamura, Hyoungeop Kim, Joo Kooi Tan, Seiji Ishikawa, Akiyoshi Yamamoto

En este artículo los autores nos presentan un algoritmo para el registro y posterior fusión de imágenes de angiografía por resonancia y tomografía computarizada para la cabeza ya que con la fusión de estas imágenes se pueden determinar más fácilmente anomalías y es de gran ayuda para el planeamiento quirúrgico y también realzar el sistema vascular para operaciones de la cabeza. Se discute que este procedimiento de registro y fusión en general es hecho manualmente y consume mucho tiempo y tiende a ser subjetivo. El método de registro propuesto es automático y de gran precisión y usan proyecciones 2D en restrictivos volúmenes de interés para mejorar el procesamiento.

MR to ultrasound registration for image-guided prostate interventions. Registro de RM y ultrasonido para intervenciones de próstata guiadas por imágenes [40].

Autores: Yipeng Hu, Hashim Uddin Ahmed, Zeike Taylor, Clare Allen^c Mark Emberton, David Hawkes, Dean Barratt

El artículo describe un método de registro deformable para el alineamiento automático de imágenes de resonancia y ultrasonido 3D transrectal para imágenes de la glándula prostática. El método de registro emplea un modelo deformable de la superficie de la glándula, derivada de imágenes de resonancia magnética (RM) y el cual es registrado automáticamente usando ultrasonido (US) 3D aplicando máxima verosimilitud del modelo dado por intensidades de voxels que representan una estimación de los vectores normales a la superficie en la frontera de la deformación de la glándula. El método es entrenado usando datos provenientes de simulaciones biomecánicas y fue probado usando datos de ocho pacientes. La media final del error realizando 100 registros de RM-US para cada paciente fue de 2.40 mm,

Multi-modal registration of speckle-tracked freehand 3D ultrasound to CT in the

lumbar spine. Registro multimodal de speckle-tracked ultrasonido a mano alzada a CT en espina lumbar [41].

Autores: Andrew Lang , Parvin Mousavi ,Sean Gill, Gabor Fichtinger, Purang Abolmaesumi.

Los autores proponen un método de registro multimodal entre ultrasonido usando la técnica speckle-tracked (usada para analizar el movimiento de las tejidos) y volúmenes de CT de la espina. El algoritmo registraba el volumen de US hacia CT creando sub volúmenes de US que consistían en una pequeña selección de todo el volumen de US, cada sub volumen contenía imágenes solapada del volumen anterior. Con este método se logra corregir algunos errores en el speckle tracking. Los autores han demostrado que han logrado registrar correctamente 4 de 5 phantom con una tasa de éxito superior a 98 % y bajar el error speckle tracking de 3mm a 2mm.

2D/3D image registration using regression learning. Registro de imágenes 2D/3D usando aprendizaje de regresión [42]

Autores: Chen-Rui Chou Brandon Frederick, Gig Mageras, Sha Chang, Stephen Pizer.

En este artículo se presenta un método de registro de proyecciones 2D y imágenes 3D llamado CLARET (Correction via Limited-Angle Residues in External BeamTherapy corrección vía residuos de ángulos limitados en terapia por radiación) el cual es un novedoso método que puede detectar rápidamente movimientos rígidos o deformaciones en objetos 3D usando proyecciones 2D de la imagen o subconjunto de esta. El método consiste en dos etapas: registro procedido por el espacio de forma y aprendizaje de regresión. En la etapa de registro, operadores lineales son usados iterativamente para estimar el movimiento/deformación basados en parámetros de la actual residuo de la intensidad entre las proyecciones del objetivo y las reconstrucciones estimadas de la imagen 3D. La otra etapa usando ejemplos de regresión en el tiempo producidas por las imágenes 3D, formula la relación entre los parámetros y la covarianza de las intensidades 2D de las proyecciones. El método es aplicado a sistemas guiados por imágenes para tratamiento con radiación y solo requiere pocos segundos para estimar la localización del tumor por medio de proyecciones rígidas 2D.

Metodología para el registro multimodal de imágenes 3D utilizando información mutua. [17]

Autor: Oscar Andrés Vélez Martínez

En este trabajo el autor propone una metodología para el registro multimodal de imágenes 3D de tomografía computarizada (CT) e imágenes de resonancia magnética (MRI). Dado que el procesamiento previo de la imagen es fundamental para mejorar el desempeño de las etapas siguientes, se realiza una etapa de filtrado utilizando filtros espaciales (media, mediana, gaussiano y unsharp), posteriormente se evalúa el desempeño de los filtros utilizando la métrica Q (Quality), el error cuadrático medio (MSE) y la relación pico señal a ruido (PSNR). Para el registro se utilizan transformaciones rígidas, no rígidas y finalmente se realiza el registro multimodal utilizando la información mutua como métrica de desempeño. Por último se mostrarán algunos artículos en los cuales el objetivo es acelerar los algoritmos para hacerlos más rápidos y eficientes.

Increasing the Automation of a 2D-3DRegistration System. Incrementando la automatización de sistemas de registro 2D-3D. [43]

Autores: Andreas Varnavas, Tom Carrell y Graeme Penney,

En el artículo se presenta 4 novedosos métodos de registro en especial referentes de vértebras, dos de las técnicas mostradas están basadas en la inserción intraoperativa de una marca fiducial en datos pre operativos, los otros dos usan una medida de similitud entre varias imágenes de vértebras de tomografía y fluoroscopia. Se discute como muchos de los algoritmos actuales se necesita una interacción manual para el registro en especial en la estimación de la pose inicial y la verificación final. Los métodos propuestos fueron evaluados usando 31 operaciones (31 tomografías y 419 imágenes de fluoroscopia) y se demostró que estos métodos pueden remover la necesidad de la identificación inicial de la vértebra y ser muy efectivo con una tasa de éxito del 100 % en la etapa de verificación.

Real-time image-based rigid registration of three-dimensional ultrasound. Registro rígido de imágenes ultrasonido en tres dimensiones en tiempo real [44].

Autores: Robert J. Schneider a, Douglas P. Perrin, Nikolay V. Vasilyev, Gerald R. Marx, Pedro J. del Nido, Robert D. Howe.

Se presenta un algoritmo de registro en tiempo real para volúmenes tridimensionales de ultrasonido (3DUS), los volúmenes son registrados tan rápidos como son adquiridos gracias al uso de un marco acelerado en unidades de procesamiento gráfico. Se comenta como este registro es necesario en muchas aplicaciones tales como cuando se desea expandir el campo visión para estabilizar temporalmente la secuencia de ultrasonido para cancelar los movimientos de la sonda. Los sistemas actuales de registro de 3DUS suelen usar sistemas externos de seguimiento óptico o electromagnético pero son caros e imponen restricciones en la adquisición y no son capaces de hacerlo en tiempo real limitando la retroalimentación hacia el médico. En este artículo el algoritmo de registro rígido basado en características es capaz de corregir los pequeños desplazamientos de la sonda cuando ocurren entre cortes y operar en tiempo real y a tenido una precisión comparable ha los métodos de registro basados en características.

Capítulo 2

Diseño de los algoritmos: pruebas e implementación

Para la realización del diseño del algoritmo se hicieron varias pruebas para cuantificar el desempeño de las diferentes partes y así escoger la mejor combinación entre métricas, transformaciones espaciales y técnicas de optimización. El sistema que sirvió para las pruebas fue AMD Athlon II P320 Dual core a 2.10GHz, con 3Gb de ram a 1333MHz.

2.1. Métricas

Para las métricas vistas en el marco teórico, se realizó una prueba inicial de cómo condiciones cambiantes en las imágenes pueden afectar a la métrica. Para esta prueba se escogió una imagen genérica, ver figura 2.1, de esa imagen se extrajo una muestra de 57×41 en un lugar específico que será la referencia, ver figura 2.2. Luego esa muestra de la imagen se comparó con varias partes de la misma imagen pero con condiciones diferentes, la idea es recorrer la imagen alterada y escoger donde detecte la mejor similitud, si ésta coincide con el lugar donde se extrajo la referencia, la técnica tendrá éxito en la prueba.



Figura 2.1: Imagen genérica para pruebas

Entre las condiciones cambiantes tenemos:



Figura 2.2: Muestra de 57×41

- set1 : La misma imagen (control)
- set2 : La misma imagen con ruido Gaussiano
- set3 : La misma imagen con ruido de Poisson
- set4 : La misma imagen con ruido impulsivo (sal y pimienta)
- set5 : La misma imagen pero negativa.
- set6 : La misma imagen con más brillo.
- set7 : La misma imagen con menos brillo.
- set8 : La misma imagen con más contraste.
- set9 : La misma imagen con menos contraste.

Entre las métricas analizadas tenemos:

- Norma L1.
- Suma cuadrado de las diferencias.
- Coeficiente de correlación.
- Radio mínimo.
- Información mutua basada en Shannon normalizada y no normalizada.
- Información mutua basada en Rényi.

Los resultados para la norma L1 se expresan en la tabla 2.1

Como se puede apreciar en la tabla 2.1 con el set1 de control el resultado la norma L1 es 0 pues las imágenes son iguales. Para los sets 5 : negativo y 7 menos brillo la técnica falló en el reconocimiento. El set 5 al ser negativo simula de cierto modo una imagen multimodal pues la información de los píxeles es opuesta y fue donde más diferencia hubo entre los píxeles.

Los resultados para la suma de la diferencia al cuadrado se expresan en la tabla 2.2

La técnica presenta resultados similares a la norma L1.

Los resultados para el coeficiente de correlación se expresan en la tabla 2.3

Norma L1	valor(pixeles)	Éxito
set1	0,0000	si
set2	107,3333	si
set3	91,0235	si
set4	62,6257	si
set5	841,6196	no
set6	241,7020	si
set7	447,4980	no
set8	257,7412	si
set9	203,9804	si

Cuadro 2.1: Tabla norma L1: valor del resultado de la norma L1 y si esta tiene éxito al encontrar al encontrar la muestra

SCD	Diferencia(pixeles)	Éxito
set1	0,00	si
set2	219,93	si
set3	157,00	si
set4	1120,90	si
set5	3981,70	no
set6	856,79	si
set7	2502,90	no
set8	1219,00	si
set9	771,10	si

Cuadro 2.2: Tabla SSD:valor del resultado de la SSD y si esta tiene éxito al encontrar al encontrar la muestra

CC	Coficiente	Éxito	Error porcentual (%)
set1	1,0000	si	0,00
set2	0,9448	si	5,52
set3	0,9598	si	4,02
set4	0,7634	si	23,66
set5	-1,0000	no	200,00
set6	1,0000	si	0,00
set7	1,0000	si	0,00
set8	0,9495	si	5,05
set9	0,9999	si	0,01

Cuadro 2.3: Tabla CC:valor del resultado del CC y si este tiene éxito al encontrar al encontrar la muestra, con su error porcentual

Según la tabla 2.3 en lo referente al coeficiente de correlación se observa un error del 4 % al 23.66 % en los ruidos empleados. En los set 6, 7 (brillo) se ve cómo el error es 0, esto es verificado en [11] y en [12], donde se afirma que existe una relación lineal entre los valores de las intensidad bajo condiciones cambiantes en la iluminación. En el set 5 (negativo) se aprecia otra propiedad del CC donde el valor es -1, esto significa que la imagen es inversa a la otra, exactamente lo que es el set5.

Los resultados para el radio mínimo se expresan en la tabla 2.4

Radio mínimo	Relación	Éxito	Error porcentual (%)
set1	1,0000	si	0,00
set2	0,9474	si	5,26
set3	0,9401	si	5,99
set4	0,9651	si	3,49
set5	0,5257	no	47,43
set6	0,8479	si	15,21
set7	0,7046	no	29,54
set8	0,8520	si	14,80
set9	0,8811	si	11,89

Cuadro 2.4: Tabla del radio mínimo: valor del resultado del radio mínimo y si este tiene éxito al encontrar al encontrar la muestra, con su error porcentual

Como se aprecia en la tabla 2.4 para el set1 (control) la relación entre intensidades es 1, osea que son iguales, en los set 2,3,4 se puede observar que la técnica se comporta de manera adecuada ante datos con ruido sobretodo para el ruido impulsivo. En los set 5 (negativo) y 7 (con más brillo) es donde la técnica falla y presenta más error. Para los demás sets, la técnica aunque tiene éxito en la búsqueda, presenta un error de alrededor del 12 % al 15 %.

Los resultados para la Información mutua basada en Shannon no normalizada IM se expresan en la tabla 2.5.

IM Shannon	IM	Error Porcentual(%)	Éxito
set1	6,8452	0	si
set2	3,5180	48,61	no
set3	3,4966	48,92	no
set4	6,5462	4,37	si
set5	6,8452	0	si
set6	6,2811	8,24	si
set7	6,2948	8,04	si
set8	5,5131	19,46	si
set9	5,9220	13,49	si

Cuadro 2.5: Información mutua de Shannon:valor del resultado de la IM y si esta tiene éxito al encontrar al encontrar la muestra, con su error porcentual

Para la información mutua basada Shannon el valor del set 1 del control expresa el valor máximo de entropía de las imágenes y sirve de comparación para los demás. En los sets 2 y 3 que son ruido Gaussiano y Poisson, la IM de Shannon falla a estos tipos de ruido como se aprecia en [12] y en [17],

pero con el set 4 ruido impulsivo solo se ve un error porcentual del 4.37 %. La característica mas importante de la técnica se observa en el set 5 (negativo), que da un error de 0, demostrando que la IM es capaz de comparar imágenes multimodales. En los demás sets referentes a la iluminación de la escena tiene un error entre un 8.24 % y un 19.46 %.

Una mejora a la IM basada en la entropía del Shannon está basado en el trabajo de Studholme, la cual sería la información mutua normalizada IMN, según [12], es una versión más robusta que la IM normal. Los resultados se expresan en la tabla 2.6

IMN Shannon	IMN	Error Porcentual(%)	Éxito
set1	2,0000	0,00	si
set2	1,3188	34,06	no
set3	1,3235	33,83	si
set4	1,9223	3,89	si
set5	1,9991	0,05	si
set6	1,9176	4,12	si
set7	1,9188	4,06	si
set8	1,8054	9,73	si
set9	1,8651	6,75	si

Cuadro 2.6: Información mutua normalizada:valor del resultado de la IMN y si esta tiene éxito al encontrar al encontrar la muestra, con su error porcentual

Si se compara la IM no normalizada 2.5 con la normalizada 2.6 se observa un error reducido, lo que la hace menos sensible a los cambios de iluminación y a los ruidos.

Los resultados para la Información mutua basada en Rényi se expresan en la tabla 2.7

IM Rényi $\alpha=2$	IM	Error Porcentual(%)	Éxito
set1	2,0007	0	si
set2	1,2825	35,90	si
set3	1,2890	35,57	si
set4	1,9524	2,41	si
set5	2,0000	0,03	si
set6	1,9023	4,92	si
set7	1,8986	5,10	si
set8	1,6358	18,23	si
set9	1,8532	7,37	si

Cuadro 2.7: Información mutua de Rényi con α 2: valor del resultado IMR y si esta tiene éxito al encontrar al encontrar la muestra, con su error porcentual

la entropía de Rényi tiene un orden α que afecta el resultado de la información mutua, para Rényi de orden 2 2.7. Ante el ruido Gaussiano y de Poisson el error porcentual fue del 35 % pero comparado con la la información mutua de Shannon 48 % tiene un error reducido. Bajo condiciones de iluminación tiene resultados parecidos pero se observa que en el set 8 (alto contraste), aunque tiene éxito el error es de 18.24 %.

Dado el estudio anterior, se demuestra que las mejores opciones para el registro de imágenes

multimodales están en la información mutua usando ya sea Shannon (normalizada o no normalizada) o Rényi, e incluso la de Tsallis, aunque no incluida en el estudio. Los demás métodos fallan en la prueba del set5 que emula una imagen multimodal. Sin embargo, ya que la entropía de Rényi depende de un factor α y que según [12] es computacionalmente más costosa que IM se ha optado por usar solo IM normalizada y no normalizada.

Para tener una comparación entre la IM normal y normalizada se preparó otra prueba, esta vez con un par de imágenes multimodales provenientes de <http://es.mathworks.com/help/images/ref/imregister.html>. Este set es una imagen de rodilla tomada con MRI con eco de spin y otra tomada con MRI eco de spin con recuperación invertida. Como se aprecia en la figura 2.3 aunque tomadas en el mismo lugar la intensidad de los pixeles es diferente. Las dos imágenes fueron alineadas para poder tener un control sobre la transformación espaciales ya que las imágenes originales vienen ligeramente desalineadas. Luego de alinearlas se roto 10 grados, así el registro solo dependió de la rotación.

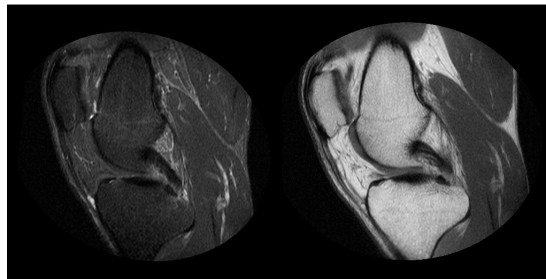


Figura 2.3: Imágenes de rodilla con MRI diferentes secuencias: izquierda con eco de spin con recuperación invertida , derecha: con eco de spin

En la figura 2.3 se observa que tanto la IM como la IMN tienen éxito al registrar el set de la figura 2.3, en pruebas mas exhaustivas realizadas en el capítulo 3.1.3 acerca de la robustez del algoritmo, se puede apreciar que la IM normalizada es más robusta que la IM no normalizada. Sin embargo, la diferencia entre los dos algoritmos es mínima por lo tanto se implementaron las dos formas en C# y Matlab.

El algoritmo empleado para la implementación del registro de imágenes multimodales consiste de los siguientes pasos :

1. Comparar las dos imágenes (referencia y objetivo) vía información mutua.
2. Establecer un transformación espacial a la imagen objetivo.
3. Comparar nuevamente las imágenes y se busca la optima transformación que maximice la información mutua entre las imágenes.
4. Fusionar la imagen de referencia y la objetivo transformada.

Para la transformación espacial se usó una transformación rígida. Aunque lo ideal es usar una transformación afín, uno de los requisitos de diseño es el de no modificar la escala de la imagen pues de ello se encarga el proceso de reconocimiento. Pero una transformación afín puede ser fácilmente implementada en el futuro. Por cuestiones de eficiencia se usaron librerías computacionales (Matlab y OpenCV) para este fin, pues ya están altamente optimizadas para reducir el tiempo de desarrollo.

2.2. Algoritmos en Matlab

El algoritmo de registro utilizado en Matlab consta de las siguientes etapas:

```
Leer imagen_referencia, imagen_objetivo
SI tamaño(imagen_referencia )= tamaño(imagen_objetivo )
Leer coordenadas_iniciales(angulo_inicial, x_inicial, y_inicial)

[angulo, x,y , mejor_valor]=Optimizar(-informacion_mutua(imagen_referencia,
    imagen_objetivo, coordenadas_iniciales)

imagen_objetivo=rotar(imagen_objetivo, angulo)
imagen_objetivo=trasladar(imagen_objetivo,x,y)
de lo contrario
Mensaje "no son de igual tamaño"
FIN SI
```

Como se necesita maximizar la función `informacion_mutua`, pero los algoritmos de optimación siempre buscan el mínimo de la función, se optimiza el negativo de la función.

La función información mutua es:

```
Leer imagen1, imagen2

Calcular histograma_conjunto(imagen1,imagen2)
Calcular hx=entropia_marginal(imagen1)
Calcular hy=entropia_marginal(imagen1)
Calcular hxy=entropia_conjunta(imagen1)
Retornar IM=hx+hy-hxy
```

Para el caso de la información mutua normalizada los pasos son los mismos solo que se retorna $IM = \frac{hx+hy}{hxy}$ en vez de $IM = hx + hy - hxy$

Para su utilización como función ha ser minimizada en donde los parámetros son (angulo,x,y) se deben tener la siguiente estructura:

```
funcion informacion_mutua(angulo , x, y)
Leer imagen1, imagen2
img_movil=imagen2;

img_movil=rotar(imagen2,angulo)
img_movil=trasladar(img_movil,x,y)

Calcular histograma_conjunto(imagen1,imagen2)
Calcular hx=entropia_marginal(imagen1)
```

```

Calcular hy=entropia_marginal(imagen1)
Calcular hxy=entropia_conjunta(imagen1)
Retornar IM=-(hx+hy-hxy)

```

En cada iteración del optimizador los parámetros (ángulo,x ,y) cambian, el optimizador busca la mejor solución que minimize la función, pero como se requiere maximizarla basta con cambiar el signo de la función.

En matlab se utilizaron las siguientes funciones. Para mas información acerca de ellas consultar el apéndice A

2.2.1. Optimizador

```
[x,eval] = fminsearch(@fun,x0)
```

Donde fun es la función a ser minimizada, x0 es un vector con las posiciones iniciales, x es un vector con las mejores posiciones encontradas y eval es el mínimo valor de la función encontrado. Fminsearch busca el mínimo de una función multivariable sin restricciones usando un método libre de derivadas.

2.2.2. Información mutua

```
MI2(image_ref,image_target)
```

Calcula la información mutua de dos imágenes, Imagen de referencia **image_ref** y la imagen objetivo **image_target**. Las dos imágenes deben tener igual tamaño, estar en escala de grises y tener una profundidad de los niveles de gris de 8 bits (0-255).

La función calcula tanto la entropía marginal de las imágenes, como su entropía conjunta. Para el calculo de la entropía conjunta es necesario calcular el histograma conjunto de las imágenes.

2.2.3. Histograma conjunto:

```
h=joint_h(image_1,image_2)
```

El objetivo de este algoritmo es extraer el histograma conjunto de un par de imágenes del mismo tamaño. Es importante recordar que la imagen deber tener una profundidad de colores de 8 bits.

2.2.4. Transformación geométrica

```
img_movil=transladar(img, x,y)
```

Aplica una transformación espacial que traslada la imagen `img` hacia una coordenada `x,y`.

```
img_movel=imrotate(img, ang,'bilinear','crop');
```

Función de matlab que rota una imagen `img`, cierto ángulo `ang`. Para el uso en el proceso de registro se utilizó el parámetro `'crop'`, para que la imagen resultante no cambie su tamaño. El parámetro `'bilinear'` fue el interpolador utilizado.

2.3. Implementación como librería del neuronavegador

El algoritmo diseñado en matlab es capaz de registrar un par de imágenes multimodales dando ciertas posiciones iniciales, pues el método de Nead Nelder que utiliza matlab es para optimizaciones locales, es decir, solo busca mínimos locales y por lo tanto es posible que la solución no sea la correcta. Para su implementación en C# el algoritmo de la información mutua utilizado es idéntico a la versión en matlab pero cambia en la forma de la transformación espacial y el optimizador, para estos fines se utilizaron dos librerías computacionales: OpenCV y liboptimization.

La librería OpenCV (Open Source Computer Vision Library) <http://opencv.org/> es una librería de código abierto para visión por computadora y aprendizaje de máquina. La librería se usó para la lectura y adquisición de las imágenes y de las transformaciones espaciales. OpenCV esta diseñado para C y C++, para su uso con el software de neuronavegación, que está escrito en C# se uso el wrap llamado Opencvsharp <https://github.com/shimat/opencvsharp>.

Para la lectura de imágenes se usó el formato `iplimage` de `Opencv`. Para la transformación espacial se usó la función `Cv.WarpAffine` que es capaz de realizar una transformación afín: trasladar `x,y`, rotar y cambiar de escala, en su forma básica:

```
Cv.WarpAffine(imag, imagen_objetivo, mapMatrix, Interpolador);
```

Donde `img` es la imagen a la que se le va a aplicar la transformación, definida en `mapMatrix` y `imagen_objetivo` donde se va a guardar el resultado.

El interpolador usado afectará el proceso de registro pues inducirá en la imagen artefactos indeseados que a su vez generarán errores en el algoritmo de comparación. Una comparación entre 4 tipos de interpolador, registrando imágenes de TAC con Angiografía usando IMN, se puede ver en la tabla 2.8. Con otros set por ejm: monomodal o algunos set multimodales, la tasa de éxito puede estar alrededor del 100 %, sin importar el método de interpolación y no se apreciaría el cambio en este. La fusión de las imágenes registradas se realiza con la función:

```
Cv.AddWeighted(img1, alpha, img2, beta, 0.0, dstimg);
```

Donde img1 y img2 son las imágenes a fusionar, alpha y beta son los pesos relativos que cada imagen tendrá en el momento de la fusión y dstimg es la imagen donde quedara guardada la fusión.

	Cubico	Lineal	Vecinos	Lancos4
dato	tiempo(ms)	tiempo(ms)	tiempo(ms)	tiempo(ms)
1	6879,9656	7564,2573	2451,4017	16338,0010
2	10670,8256	7120,5859	5768,8882	17650,8237
3	6268,4231	5795,8017	6087,3787	16333,4912
4	3766,7149	5761,4472	4424,6212	21636,1255
5	9413,5448	4882,7464	4471,2996	20338,4360
6	9383,8157	3595,2926	3234,2954	8130,0912
7	6908,0760	4508,0429	4836,4948	14735,5451
8	7514,7408	6221,9657	4029,2955	9436,7207
9	6256,5431	6684,7709	2801,8402	13453,5795
10	4378,2138	4482,0225	6029,6871	16154,6333
11	13817,2608	2642,7873	4842,3382	21613,8251
12	9384,2545	7132,7905	6432,4107	17625,6406
13	9355,7359	4432,0446	2378,8985	16141,7312
14	3773,9495	6291,7095	3604,7129	16222,9278
15	11286,8651	6291,2401	4459,6468	8108,4077
promedio	7937,2619	5560,5003	4390,2140	15594,6653
Confiabilidad	87 %	73 %	87 %	53 %

Cuadro 2.8: Interpoladores usados: cúbico o bicúbico, Lineal, Vecinos mas cercanos, Lancos4

En la tabla 2.8 se observa que los mejores métodos son el cúbico y vecinos más cercanos, pero en una inspección visual sobre el registro se consideró que es mucho más preciso el cúbico, sin embargo el tiempo de cómputo es mucho mayor. Se deja la opción de cambiarlo en el proceso de registro según las necesidades.

Se crearon dos funciones para realizar una transformación espacial; una rígida (traslación y rotación) y otra afin (traslación y rotación). Sin embargo como el neuronavegador puede tener la capacidad de escalar las imágenes usando un volumen de control, se implemento en la mayoría de las pruebas una transformación rígida, expresada en:

```
public static IplImage rigida(IplImage img, double x, double y, double ang)
```

La función `rigida` toma una imagen en el formato `IplImage` de `opencv` y la traslada un valor `x,y` y la rota un ángulo `ang`.

Sin embargo también se creo una función para realizar una traslación afin:

```
public static IplImage afin(IplImage img, double x, double y, double ang, double esca)
```

Esta función se comporta de manera similar a la rígida, pero incluye un valor para cambiar la escala de la imagen.

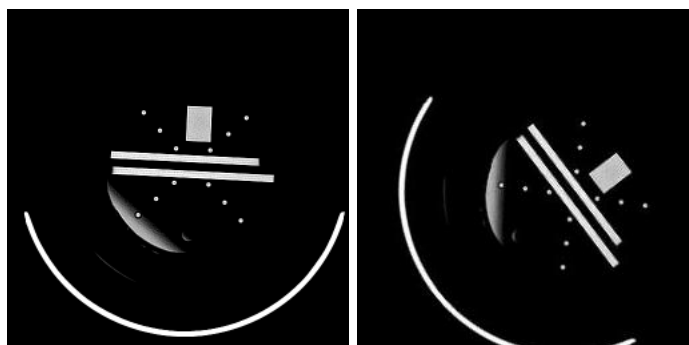
LibOptimization: se encarga de los modelos de optimización para la búsqueda de la mejor solución al problema del registro. `LibOptimization` es una librería con algoritmos de optimización

para .NET, que tiene implementado varios métodos tales como gradiente descendente, método de Newton, método de Nelder Mean, algoritmos genéticos, de enjambre , evolución diferencial entre otros. LibOptimization puede obtenerse de <https://github.com/tomitomi3/LibOptimization>

Entre los modelos probados se encuentran:

- Standard Cuckoo search (CS): Búsqueda Cuco
- Differential Evolution algorithm (DE): Evolución diferencial
- Basic Particle Swarm Optimization (PSO): Optimización por enjambre de partículas con varias variantes PSOAIW (adaptative inertial weight) PSOCIW(Chaotic Inertial weight) PSOLDW (Linear decrease weight)

A continuación se verá una prueba del tiempo de cómputo y la confiabilidad para registrar una imagen proveniente de TAC, ver figuras 2.4a y 2.4b, que ha sido rotada 50 y trasladada 20 pixeles en x y 10 pixeles en y. Para examinar la confiabilidad se tomaron 10 datos y se observó la precisión, se tendrá por éxito en el registro si la imagen registrada no excede los 2 grados en rotación o 2 pixeles en x o en y. Si no cumple con esta condición la imagen se considera como desalineada y por lo tanto sin éxito en el registro. Se debe tener en cuenta que el factor de precisión de estos algoritmos está determinada por las necesidades del sistema y que varían de acuerdo con este. Para el tiempo de cómputo se tomaron 5 valores. Estas pruebas sirvieron para escoger el mejor algoritmo y hacer pruebas más extensas sobre el. Fueron probadas usando la IM no normalizada.



(a) Imagen de RM del Phantom (b) Imagen rotada 50 grados y trasladada x=20 y=10

Figura 2.4: Set de Imágenes monomodal

Cada algoritmo de optimización tiene sus distintivas características y configuraciones con lo que es imposible realizar una prueba bajo los mismos parámetros para todos por igual, además estos parámetros tienen diferentes efectos de acuerdo al algoritmo empleado, por ejemplo cierto algoritmo realizará más acciones en una iteración que otro y por lo tanto necesitará más tiempo para su ejecución, pero convergerá más rápido, por lo tanto dicho algoritmo necesitará menos iteraciones que otro.

Los parámetros del algoritmo fueron escogidos teniendo en cuenta dos aspectos: primero que el tiempo para encontrar la solución fueran lo mínimo posible y que la confiabilidad del algoritmo

estuviera por encima del 70 % . Entre los parámetros analizados tenemos: valor de criterio de parada EPS (Epsilon), el número de iteraciones (I) y la población inicial en el espacio de búsqueda (PI). Los parámetros por defecto del algoritmo dan un tiempo de ejecución muy extensos por encima de los 10 minutos, por lo tanto no prácticos.

En la prueba se configuró los parámetros antes mencionados de tal manera que el tiempo de ejecución fuera el menos posible dentro de los alcances de dicha prueba, el único común entre ellos fue EPS 1.7E-3. Para los basados en PSO la PI fue de 40 y las I fueron 20, para CS la PI fue de 10 y las I fueron 60, para DE PI=40,I 40 como se ve en la tabla 2.9.

Algoritmo	PSOAIW	PSOCIW	PSOLDIW	DE	CS
Confiabilidad(%)	100 %	100 %	90 %	100 %	90 %
	tiempo(ms)	tiempo(ms)	tiempo(ms)	tiempo(ms)	tiempo(ms)
	10885,6709	11036,2421	11480,9337	37152,4444	29758,3535
	10892,8832	11123,0557	11096,1874	36982,7211	29824,8570
	11026,3311	11056,2961	11189,6202	37083,4640	30192,7610
	11050,8856	10931,2987	11051,7660	36830,6443	29897,6473
	10888,5570	11014,7554	11170,5689	36788,1797	29732,0107
promedio	10948,8656	11032,3296	11197,8152	36967,4907	29881,1259

Cuadro 2.9: Tiempo de computo Algoritmos de optimización PSOAIW, PSOCIW, PSOLDW,DE,CS

En la tabla 2.9 se ha considerado que el mejor algoritmo es PSOAIW, pero como las pruebas realizadas tenían pocos datos, se consideró una prueba mas grande con 30 datos 2.10 donde se observa una confiabilidad del 100 % con un tiempo promedio de 10959,5053 ms, también se observa el error entre el resultado de las coordenadas de registro: ángulo θ (50 grados) y (x,y) x=20 , y =10.

Los resultados de la tabla 2.10 fueron realizados usando la IM no normalizada, para la version normalizada ver 2.11 se observa un tiempo promedio de 10415.2129 ms la cual comparada con la IM normal de 10959.5053 ms tubo una reducción del tiempo de cómputo de 544,2924 ms, alrededor de un 5 % mas rápido. Si consideramos los algoritmos, la diferencia entre la IM normal y la normalizada es poca ya que la normal es una suma de entropías, en la IM normalizada es una suma y una división, la reducción en el tiempo sigue demostrando la robustez de IM normalizada frente a la IM normal.

Con la información anterior se concluye que el mejor método para el registro de imágenes multimodales es el uso de la información mutua normalizada y algoritmo de optimización por enjambre de partículas con peso inercial adaptativo PSOAIW, con interpolador cúbico.

2.3.1. Preproceso en las imágenes adquiridas

Por lo general las imágenes provenientes de TAC o RM, al ser pre operatorias, se puede acceder fácilmente al archivo DICOM y a la información contenida en dicho archivo, por lo que generalmente no se necesita ningún preproceso, por ejemplo, ver la imágenes 2.5b,2.5a. Sin embargo en las imágenes de angiografía y ultrasonido se requiere en ocasiones de una forma de eliminar partes indeseables en las imágenes, como se ven en las figuras 2.6a, 2.6b, donde en la imagen de ultraso-

dato	tiempo(ms)	angulo(grado)	error	x(pixeles)	error	y(pixeles)	error
1	10885,6709	49,97	0,03	19,49	0,51	10,20	0,2
2	10892,8832	49,95	0,05	19,33	0,67	10,17	0,17
3	8621,0111	49,67	0,33	19,44	0,56	10,29	0,29
4	10911,5967	49,98	0,02	19,39	0,61	10,19	0,19
5	10936,1259	50,03	0,03	19,62	0,38	10,22	0,22
6	11020,3311	50,04	0,04	19,16	0,84	10,24	0,24
7	10892,8093	49,94	0,06	19,56	0,44	10,11	0,11
8	11073,7206	50,15	0,15	19,41	0,59	10,04	0,04
9	11133,3525	49,99	0,01	19,39	0,61	10,03	0,03
10	10876,5659	50,05	0,05	19,26	0,74	10,22	0,22
11	11066,1368	49,78	0,22	19,41	0,59	10,13	0,13
12	10900,5266	50,04	0,04	19,43	0,57	10,21	0,21
13	10961,7604	50,06	0,06	19,52	0,48	10,11	0,11
14	10976,6240	49,97	0,03	19,43	0,57	10,24	0,24
15	11368,2833	50,01	0,01	19,45	0,55	10,25	0,25
16	10919,2724	50,09	0,09	19,43	0,57	9,98	0,02
17	10917,5540	50,06	0,06	19,39	0,61	10,15	0,15
18	10907,5692	50,05	0,05	19,55	0,45	10,23	0,23
19	10920,7873	49,97	0,03	19,43	0,57	10,25	0,25
20	10842,4707	49,99	0,01	19,49	0,51	10,02	0,02
21	10886,6061	50,02	0,02	19,48	0,52	10,18	0,18
22	10858,7743	49,99	0,01	19,41	0,59	10,15	0,15
23	8607,4714	50,37	0,37	19,48	0,52	9,91	0,09
24	11091,4222	50,23	0,23	19,35	0,65	9,64	0,36
25	11011,4475	50,15	0,15	19,43	0,57	10,21	0,21
26	10901,2505	49,92	0,08	19,43	0,57	10,14	0,14
27	10868,5641	50,04	0,04	19,42	0,58	10,16	0,16
28	10982,5696	50,03	0,03	19,42	0,58	9,90	0,1
29	10888,5570	50,01	0,01	19,48	0,52	10,19	0,19
30	10871,2992	50,05	0,05	19,27	0,73	10,21	0,21
promedio	10959,5053	50,00	0,07	19,43	0,57	10,14	0,18

Cuadro 2.10: Algoritmo PSOAIW

IMN	Tiempo (ms)
1	10366,8554
2	10729,4483
3	10399,8405
4	10495,0170
5	10519,3710
6	10504,3528
7	10369,6623
8	10325,0108
9	10389,9900
10	10316,5109
11	10247,8766
12	10303,1823
13	10319,1335
14	10435,3494
15	10506,5921
promedio	10415,2129

Cuadro 2.11: Optimización con Información mutua normalizada

nido se ven datos del estudio que interfieren en el proceso y en la imagen de angiografía se observa un fondo recortado arriba y abajo, estos artefactos afectarán al proceso de registro, no permitiendo tener éxito en este.

Para solucionar los inconvenientes en las imágenes se recurre a recortar de la imagen los artefactos indeseados, como se ve en las figuras 2.7a y 2.7b. Aunque se puede utilizar algún software de edición de imágenes para dicho fin, se prepararon dos funciones que sirven para recortar partes de la imagen, en las funciones:

```
imcortar(im,x,y,tam)
imcortar_cir(im,x,y,tam)
```

La función `imcortar` recorta un cuadrado en el punto `x,y` de tamaño `tam`. La función `imcortar_cir` recorta un círculo en el punto `x,y` de tamaño `tam`. En las imágenes de US, existen algunos tag del archivo DICOM que facilitan identificar zonas en la imagen, sin embargo no están disponibles en todos los modelos.

En la sección 3.1.3, se puede observar cómo afecta al ruido la IM y la IMN. Es recomendable una etapa de reducción de ruido aunque no es explícitamente necesaria. En los trabajos realizados por [17], se puede observar otras pruebas referentes al ruido sobre los procesos de registro.

2.3.2. Adquisición de imágenes

Para la adquisición de las imágenes de TAC del Phantom se utilizó un tomógrafo Phillips Brilliance CT 64 cortes.

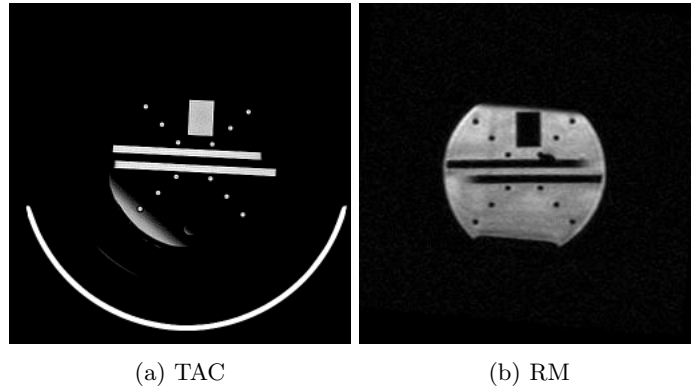


Figura 2.5: Ejemplo de estudios provenientes de TAC (a) y RM(b)

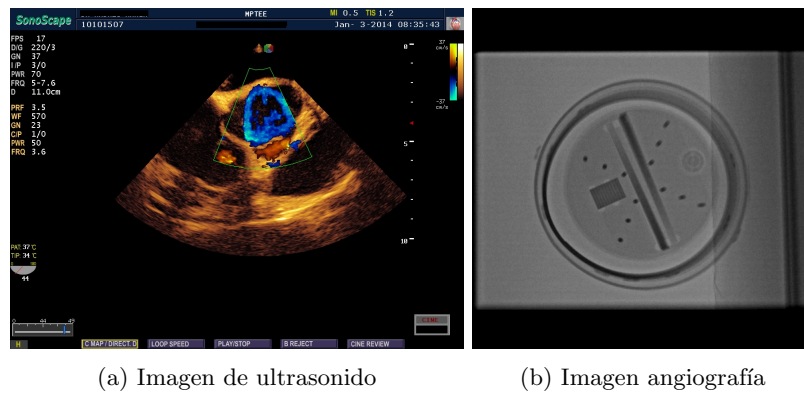


Figura 2.6: Ejemplo de estudios provenientes de US (a) y Ax(b) sin editar

Para la adquisición de las imágenes de resonancia del Phantom se utilizó un RESONADOR MAGNÉTICO SIGNA EXCITE HD GENERAL ELECTRIC.

Procedimiento:

- Parámetros de obtención de imágenes: Seleccione modo 3D, plano oblicuo y familia “Gradiente Echo”.
- Tiempo de exploración: GRE: Seleccione 1 eco. Introduzca los valores de TR, TE y del ángulo de giro para el contraste de la imagen deseada.
 - Valores de sincronización habituales en 3D: TE = mínimo, TR = 40-60, FlipAngle = 5-10, Ancho de banda de recepción = 9.
 - El valor de TE determina si la grasa y el agua están en fase o fuera de fase.
- Rango de exploración: Introduzca el FOV, el espesor de corte y los valores de espaciado de la resolución espacial y la SNR deseadas. Valores habituales de la cabeza en 3D: FOV = 16, Espesor del corte = 2, Ubicaciones = 28-68, número de secciones = 1.

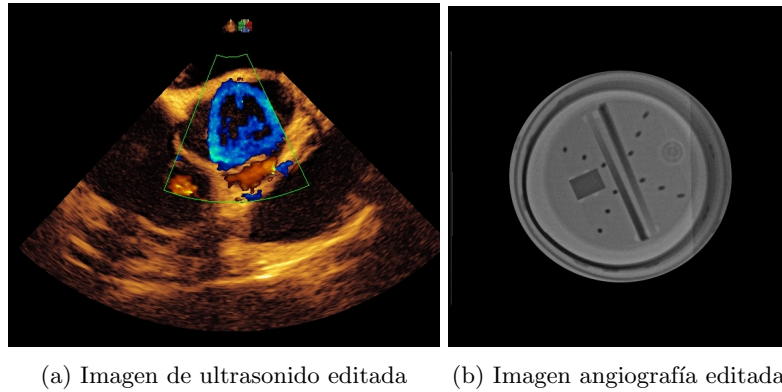


Figura 2.7: Ejemplo de estudios provenientes de US (a) y Ax(b) editados

- Tiempo de adquisición: Introduzca los valores de matriz, NEX y PFOV de la resolución espacial deseada, la SNR y el tiempo de exploración. Valores habituales de la cabeza: Frecuencia = 256, fase = 128, NEX = 1, para 3D.

Para la adquisición de las imágenes de angiografía del Phantom se utilizó un angiógrafo Siemens AXIOM Artis.

El phantom utilizado no pudo ser utilizado para ultrasonido. Se hizo una prueba con gel balístico y un equipo de ultrasonido SonoScape S8, se sumergió el phantom en dicho gel y se tomaron algunas imágenes pero no fue posible obtener una imagen nítida, por lo tanto fue descartado.

Un diseño de cómo sería el proceso de adquisición para hacer el registro, por parte del neurona-vegador utilizado por el grupo Applied Neuroscience se ve en las imágenes 2.8 y 2.9. Para el caso del angiógrafo se captura una imagen usando un volumen de control que reconoce el sistema de visión estéreo, este reconoce el punto de vista de las imágenes adquiridas por el angiógrafo, luego se utiliza los algoritmos diseñados para registrar las imágenes de angiografía con la proyección de los volúmenes ya sea de TAC o RM. Para ultrasonido se coloca una marca geométrica que reconoce el sistema, esta marca le dará información de la orientación y posición de la sonda de ultrasonido, usando un volumen de control para su inicialización, con la información de la posición, orientación y la escala de la imagen de US se procede al registro usando los algoritmos diseñados.

2.3.3. Comunicación DICOM

Como era de esperar para poder comunicarse con un aparato proveedor de imágenes diagnósticas, es necesario la creación de un servidor DICOM, el cual pueda almacenar y acceder a los imágenes de estos dispositivos. Para lograr esta comunicación con los dispositivos de imágenes médicas fue necesario el uso de 3 herramientas computacionales: La librería GDCM, VDTK y el servidor DICOM Orthanc.

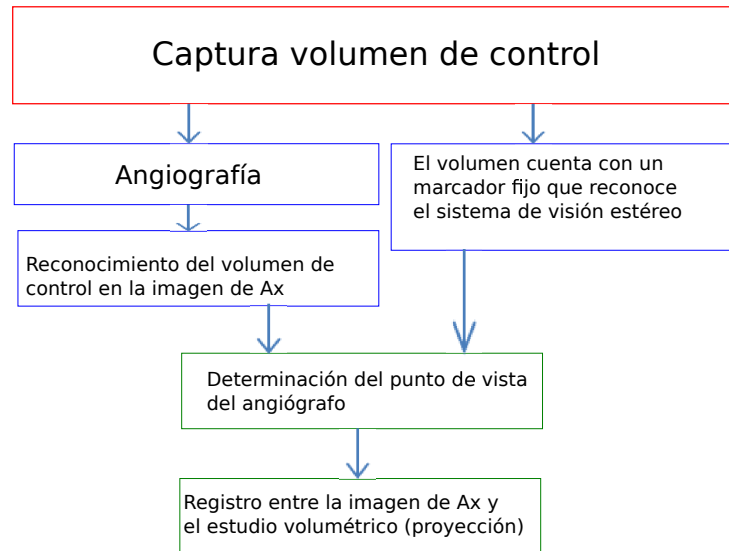


Figura 2.8: Diseño de la etapa de adquisición y registro con el angiógrafo

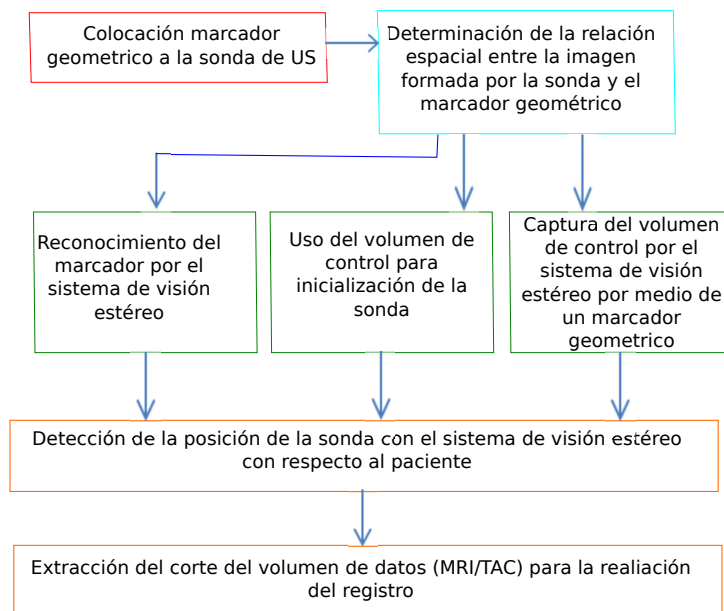


Figura 2.9: Diseño de la etapa de adquisición con equipos de ultrasonido

Librería GDCM

Grassroots DICOM (GDCM) es una librería de código abierto hecha en C++ para la implementación del estándar DICOM. Ofrece un wrapping a varios lenguajes de programación, como se ve en apéndice B. Esta librería se usó principalmente para proveer algunos comandos DIMSE como C-FIND y C-MOVE.

VDTK

DVTK es un proyecto de código abierto para probar, validar y diagnosticar protocolos de comunicación en ambientes médicos. Los detalles de la instalación y la configuración están descritos en el manual.

Cabe aclarar que la VDTK se usó para probar las distintas fases de la comunicación DICOM, los 3 módulos usado emulan en cierto modo la operación de un servidor DICOM. Entre los módulos empleados tenemos:

- Query Retrieve SCP Emulator: Este modulo sirvió para la creación de una pequeña base de datos con algunas imágenes médicas. Con el software fue posible probar lo referente a las consultas hechas con el comando C-FIND y C-MOVE creado en GDCM.
- Storage SCP emulator: Este módulo cumple con la tarea de emular un servidor de guardado. Con el se pudo probar el funcionamiento del comando C-MOVE.
- DICOM Network Analyzer: Módulo usado para capturar y diagnosticar los mensajes DICOM en las pruebas realizadas entre los componentes de la comunicación

Servidor DICOM Orthanc

Orthanc es un servidor DICOM de código abierto que provee un simple pero a la vez poderoso servidor DICOM independiente. Está diseñado para mejorar el flujo de información DICOM en hospitales y centros de investigación. Dado a la fácil configuración que esta descrita en el manual B.3, se escogió para las tareas de comunicación entre los dispositivos DICOM.

Capítulo 3

Validación, resultados y conclusiones

3.1. Validación del registro

En este capítulo se mostrarán los resultados del algoritmo de registro basado en información mutua basada en la entropía de Shannon (normalizada y no normalizada), se debe aclarar que dicho algoritmo aun no esta implementado para realizar pruebas médicas pues es un prototipo, por lo tanto tendrá varios parámetros a mejorar en estudios posteriores. El sistema que sirvió para las pruebas fue AMD Athlon II P320 Dual core a 2.10GHz, con 3Gb de ram a 1333MHz.

3.1.1. Precisión

En el capítulo 2 se mostró que el algoritmo de registro implementado es capaz de registrar con éxito un par de imágenes monomodales, pero una de las motivaciones para el desarrollo de esta investigación era el poder registrar dos imágenes multimodales preoperatorias (MRI o TAC) e intraoperatorias (Angiografía o ultrasonido), para este fin se analizó la precisión al registrar una imagen TAC con una imagen de Angiografía de un phantom figura 3.1. Dado que las imágenes producidas por el Angiografo 3.2b es una proyección y las imágenes de TAC son una serie de cortes, se creó un programa que saca a partir de todos los cortes del TAC una proyección, ver figura 3.2a.

En la tabla 3.1 se observa las distancias medidas de algunos puntos del Phantom tomados como referencia para medir, ver figura 3.3a. Con un error promedio en las distancias medidas en pixeles de D1 de 3,6 en D2 =6,7 y D3=5,1 con un error promedio en el angulo de 2,48 grados. Si usamos el tag DICOM 0028,0030 se puede convertir los pixeles de la imagen en milímetros, para este caso dicho tag vale 0.8984, así la distancia promedio es D1=3.2 mm , D2=6,0 mm , D3=4.5 mm, por el momento el software en su versión prototipo donde el mejor escenario para la aplicación del algoritmo propuesto es en la etapa de planeación quirúrgica, en donde se registran imágenes prequirúrgica. Las medidas fueron tomadas usando el software Gimp2.

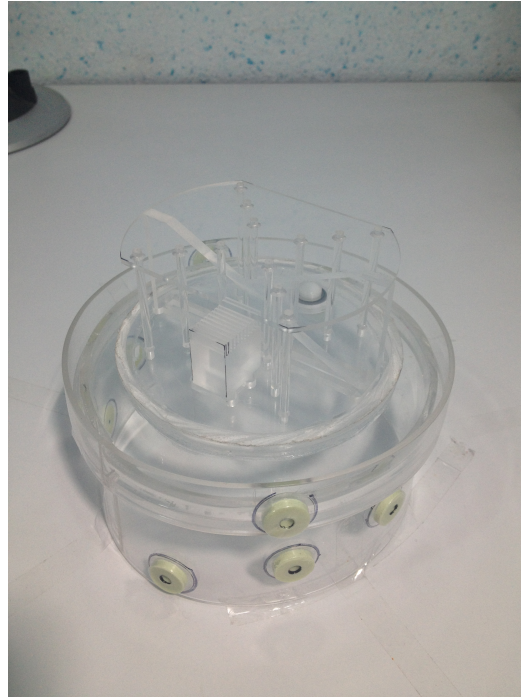
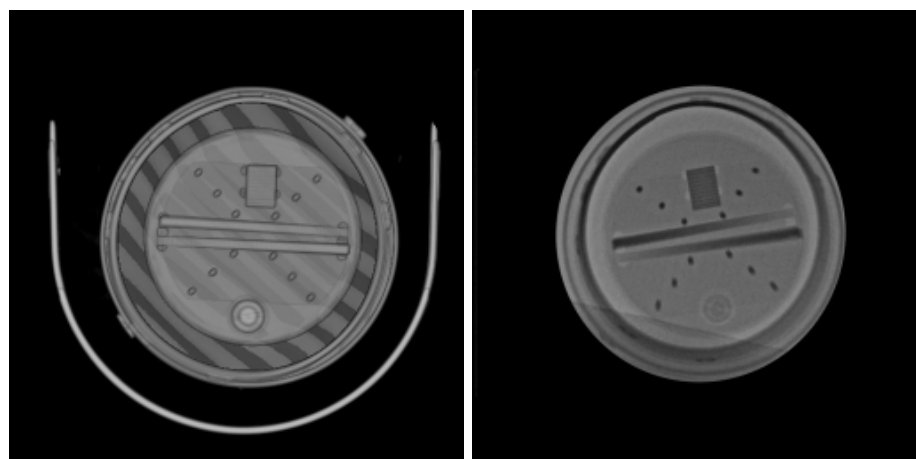


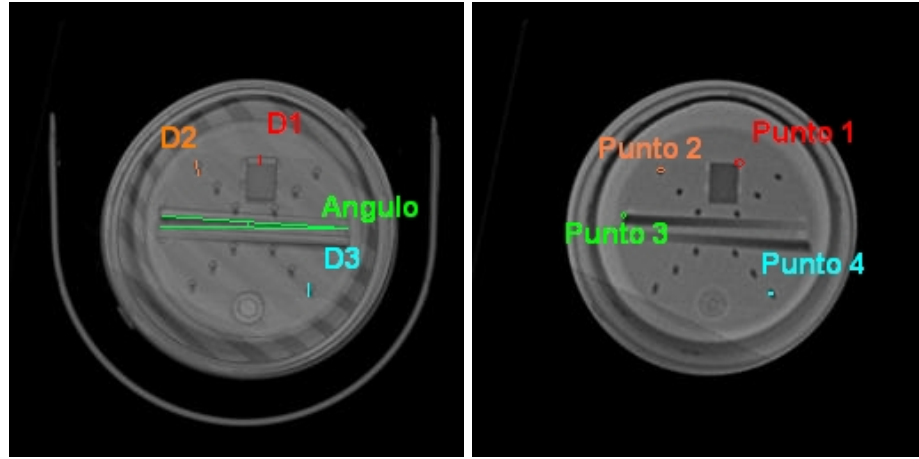
Figura 3.1: Phantom utilizado por el Grupo Appied Neuroscience



(a) Proyección del Phantom TAC

(b) Imagen del Phantom Angiógrafo

Figura 3.2: Imágenes del Phantom en TAC (a) y Ax (b)



(a) Distancias tomadas

(b) Puntos tomados

Figura 3.3: Distancias y puntos tomados en el Phantom, a: Distancias tomadas en el registro y fusión de las imágenes de angiografía y TAC. b: Puntos tomados en el Phantom en la imagen registrada de 3.2b

Imágenes	D1 (pixeles)	D2 (pixeles)	D3 (pixeles)	Angulo(grados)
a	4	7	5	2,18
b	3	5	3	3,19
c	3	4	5	2,17
d	4	7	5	3,21
e	3	7	5	2,17
f	4	7	6	1,64
g	3	9	5	2,70
h	4	7	6	2,17
i	4	7	5	2,16
j	4	7	6	3,24
error promedio	3,6	6,7	5,1	2,48
error RMS	3,6	6,8	5,2	2,54
error mínimo	3	4	5	1,64
error máximo	4	9	6	3,24

Cuadro 3.1: Precisión del algoritmo con un imagen de TAC con Angiografía: D1=distancia 1, D2= distancia 2, D3= distancia 3, Angulo

En las tablas 3.2 y 3.3 se muestra la diferencia en las coordenadas de los punto 1,2,3,4, entre la imagen de referencia TAC (puntos reales) con la imagen de Angiografía, ver figura 3.3b. Se observa poca variación en X del orden del 0,3 % y una variación mayor en Y del orden del 3,5 %. Un ejemplo del registro y la fusión de las imágenes de TAC y angiografía se ve en 3.4.

	Punto 1	Punto 1	punto2	Punto 2
	x(pixeles)	y(pixeles)	x(pixeles)	y(pixeles)
a	150	91	107	94
b	148	90	105	96
c	150	90	106	96
d	150	91	106	95
e	150	90	106	94
f	148	89	105	95
g	149	92	105	95
h	150	90	106	94
i	150	92	107	94
j	149	91	106	96
promedio	149,4	90,6	105,9	94,9
puntos reales	150,0	87,0	106,0	91,0
error	0,6	3,6	0,1	3,9
error porcentual(%)	0,4	4,1	0,1	4,3
Desviación	0,8	1,0	0,7	0,9

Cuadro 3.2: Precisión del algoritmo con un imagen de TAC con Angiografía: Diferencias en las coordenadas X,Y de los puntos 1 y 2

Para el set monomodal descrito el las figuras 2.4a y 2.4b, la cual fue rotada 50 grados , trasladada en $x=20$, $y=10$, los valores medidos con su error son: $50,00 \pm 0,12$ grados, en x $19,43 \pm 0,09$ pixeles , en y $10,14 \pm 0,13$ pixeles. Con una inspección visual de las imágenes registradas no se aprecia desalineamiento en las imágenes.

Para el set multimodal de la figura 2.3, se probaron dos tipo de transformaciones espaciales (rígida y de similitud), en la imagen se puede apreciar cómo están desalineadas originalmente, ver 3.5. Se procedió a una inspección visual del registro comparándola con el resultado del comando `imregister` de matlab. En las figuras 3.6a, 3.6b, 3.7a y 3.7b, se observan resultados similares entre los dos algoritmos para una transformación rígida y de similitud. El comando `imregister` de Matlab hace uso también de la información mutua de Shannon y un tipo de optimizador evolutivo. Los dos algoritmos tienen configuraciones por defecto.

Por último se aprecia un registro y fusión de las imágenes de la figura 3.5, con una transformación rígida pero previamente escalada e inclinada. No se aprecian desalineamientos aparentes, la fusión se obtiene con el comando de `opencv addWeighted`.

3.1.2. Confiabilidad y tiempo de computo

Para medida de la confiabilidad en el registro se usaron varios sets de imágenes: set1: imagen multimodal (Angiografía y TAC), ver imágenes 3.2b, 3.2a, set2: imagen multimodal (MRI), ver

	Punto 3	Punto3	Punto 4	Punto 4
Imagen	x(pixeles)	y(pixeles)	x(pixeles)	y(pixeles)
a	169	169	85	120
b	172	162	86	122
c	169	164	85	122
d	170	163	86	120
e	170	163	85	119
f	170	162	85	121
g	170	164	86	121
h	169	164	85	119
i	168	164	85	120
j	170	163	85	121
promedio	169,7	163,8	85,3	120,5
puntos reales	170,0	161,0	85,0	116,0
error	0,3	2,8	0,3	4,5
error porcentual(%)	0,2	1,7	0,4	3,9
Desviación	1,1	2,0	0,5	1,1

Cuadro 3.3: Precisión del algoritmo con un imagen de TAC con Angiografía: Diferencias en las coordenadas X,Y de los puntos 3 y 4

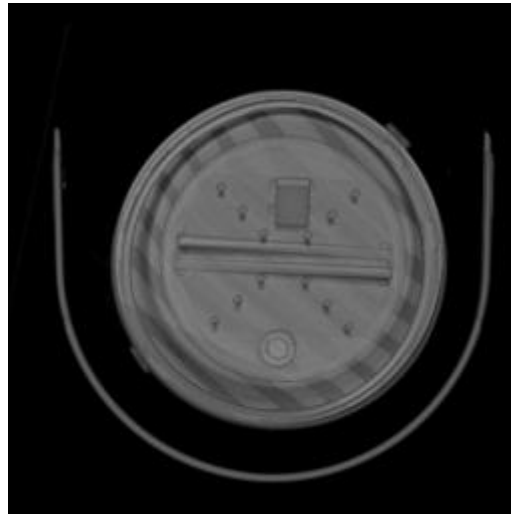


Figura 3.4: Ejemplo de registro y fusion de una imagen de angiografía y una de TAC provenientes del phantom de la figura 3.1,

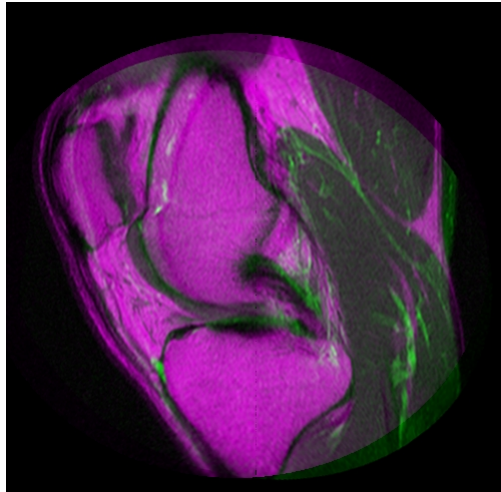
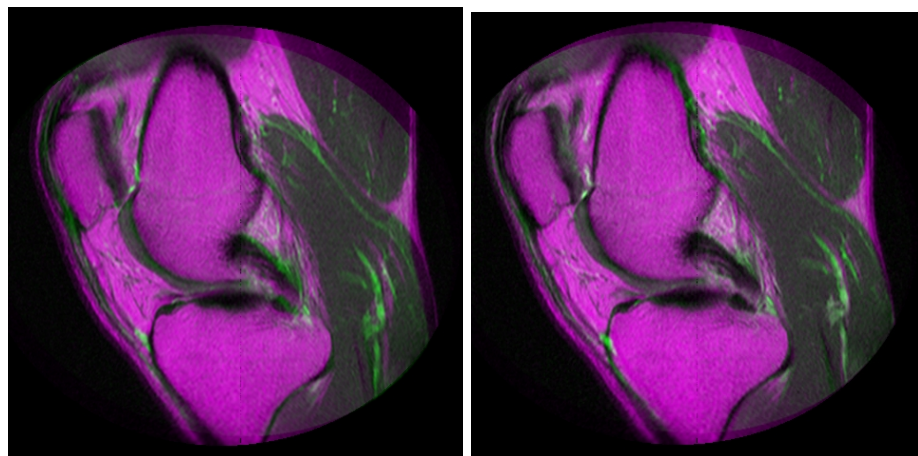


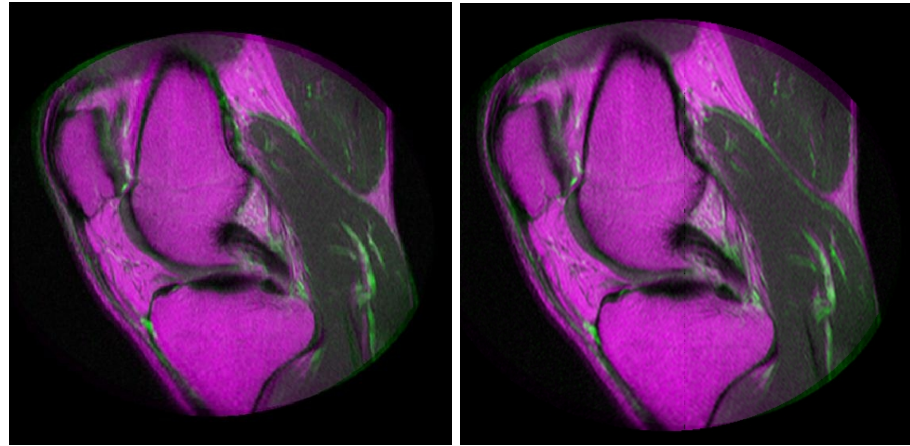
Figura 3.5: Set de imágenes multimodales fusionadas pero no registradas de la figura 2.3



(a) Matlab

(b) Algoritmo diseñado

Figura 3.6: Registro usando: a) Comando imregister matlab transformación rígida. b) Registro del algoritmo diseñado con transformación rígida



(a) Matlab

(b) Algoritmo diseñado

Figura 3.7: Registro usando: a) Comando imregister matlab transformación de similitud. b) Registro del algoritmo diseñado con transformación de similitud

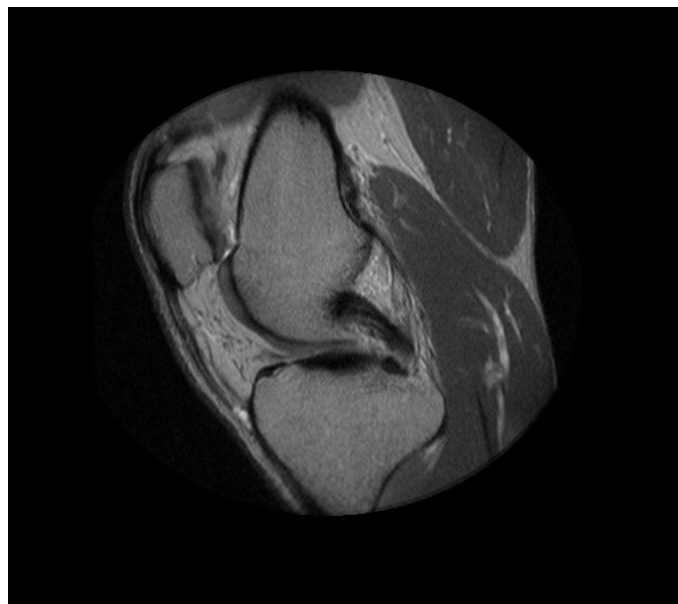


Figura 3.8: Registro y fusión de las imágenes 3.5: la imagen fue previamente escalada e inclinada

2.3, set3 el mismo set1 pero con ajuste previo de escala e inclinación y set4 monomodal, ver figuras 2.4b y 2.4a.

Con el set1, ver 3.4, se observa una confiabilidad del orden el 93 % y un tiempo de cómputo de $16161,0765 \text{ ms} \pm 948.1103 \text{ ms}$, el registro fue realizado con IMN con interpolador cúbico y algoritmo de optimización PSOAIW con epsilon $1.7\text{E-}04$. Los tipos de imágenes multimodales provenientes de angiografía o ultrasonido son los más difíciles de registrar con éxito pues las imágenes tienen muchos cambios en su estructura lo que dificulta su registro.

dato	tiempo(ms)	Angulo(grados)	X	Y	éxito
1	20795,0880	-12,77	6,66	-0,22	si
2	20280,6145	-11,89	5,62	-0,35	si
3	21409,9622	11,82	5,71	-0,35	si
4	12556,0709	-11,52	6,29	-0,50	si
5	14709,2979	-11,81	5,68	-0,38	si
6	21416,0687	-11,78	5,70	-0,35	si
7	15720,2678	11,66	5,61	4,13	no
8	24601,4866	-8,35	6,63	0,70	si
9	12643,7376	-11,59	5,62	0,43	si
10	19559,0078	-11,82	5,71	0,34	si
11	6281,0007	-12,94	9,85	0,70	si
12	17025,6073	-12,04	5,66	-0,37	si
13	15900,5090	-11,68	5,58	-0,41	si
14	18868,6057	8,68	5,72	3,25	no
15	5666,0057	-9,97	6,80	0,54	si
16	20313,8226	-10,24	5,89	-0,23	si
17	20787,4755	-12,00	5,67	0,66	si
18	18884,6249	-11,62	5,88	0,38	si
19	17586,8222	-12,08	5,75	-0,71	si
20	6254,0446	-10,84	5,98	0,01	si
21	14450,6801	-11,68	5,81	-0,33	si
22	13138,8104	-14,07	5,89	0,09	si
23	19440,8792	-12,11	5,59	-0,31	si
24	8767,1108	-11,58	5,92	-0,30	si
25	7550,2876	-8,45	6,02	0,74	si
26	18828,3836	-10,74	5,70	-0,31	si
27	18808,9703	-11,90	6,60	1,25	si
28	22533,9347	-12,11	5,58	-0,31	si
29	16881,8633	-10,35	6,62	0,61	si
30	13171,2550	-11,49	5,60	-0,62	si
promedio	16161,0765	-10,63	6,07	0,01	C=93 %

Cuadro 3.4: Confiabilidad: set TAC, angiografía con una confiabilidad C=93 % usando IMN, optimizador: PSOAIW con epsilon $=1.7\text{E-}4$ e interpolador:cúbico, Se observó si tiene éxito o no en el registro.

Para set2, ver tabla 3.5, se observa una confiabilidad del 93 % con un tiempo de cómputo promedio de $16834,7450 \text{ ms} \pm 1014,9572 \text{ ms}$. Para el set3, ver tabla 3.5, se observa una confiabilidad del 100 % con un tiempo de cómputo promedio de $22021.2148,7450 \text{ ms} \pm 1111,1982 \text{ ms}$. Ambos se

realizaron con transformación rígida e iguales condiciones en el optimizador, sin embargo el set3 tiene un tamaño de 256×256 , mientras que el set2 tiene un tamaño de 512×512 , pero el set3 de tamaño mas pequeño se demoró más que el set2 de tamaño mayor, acá se observa que el tiempo de computo no depende del tamaño de las imágenes, sino de las imágenes como tal. Aunque el set2 y el 3 tienen las mismas imágenes, el set3 es una versión escalada del set2, con lo que se necesitó de algún tipo de interpolación, este proceso pudo inducir la creación de mas máximos locales que hacen que el optimizador se demore más tiempo en encontrar la solución óptima.

	Set2		Set3	
	Tiempo(ms)	Exito	Tiempo(ms)	Éxito
1	20362,6612	Si	17571,9866	Si
2	22737,7628	Si	20822,648	Si
3	15215,8213	Si	22753,9521	Si
4	19844,5780	Si	28516,2324	Si
5	12120,0640	Si	22761,0562	Si
6	8036,0143	No	20864,5453	Si
7	18722,9033	Si	23513,5757	Si
8	14595,8546	Si	11402,1142	Si
9	18401,2400	Si	20707,6216	Si
10	16272,0569	Si	26626,7228	Si
11	15841,8915	Si	24590,2644	Si
12	13914,8884	Si	18968,6304	Si
13	21168,8775	Si	26574,1445	Si
14	18451,8173	Si	22623,5136	Si
Promedio	16834,7450	C=93 %	22021,2148	C=100 %

Cuadro 3.5: Confiabilidad C y tiempo de computo sets 2 y 3, se observo si tiene éxito o no en el registro.

Para el set4: monomodal, una prueba de confiabilidad se realizó en la tabla 2.10 sección 2.3, con un IM, algoritmo de optimización PSOAIW con Epsilon $1.7E-3$ y interpolador lineal: con una confiabilidad del 100 % y un tiempo de cómputo de $10871,2992 \text{ ms} \pm 110,2257 \text{ ms}$.

Los resultados acerca de la confiabilidad y tiempo de cómputo del algoritmo dependerán de varios factores como:

- El proceso de optimización: la configuración que se aplique ha este, afectará tanto el tiempo de procesamiento como la confiabilidad, pocas iteraciones pueden reducir el tiempo de cómputo pero provocará una mayor cantidad de fallas; una población inicial muy grande puede mejorar la confiabilidad pero aumenta el tiempo de procesamiento; un epsilon muy pequeño podrá mejorar el registro pero aumentando considerablemente el tiempo de procesamiento, entre otros.
- El tipo de transformación, ya sea de similitud o rígida, afectara el tiempo de procesamiento y los parámetros en el optimizador que deben ser diferentes ya sea para una o la otra. La transformacion de similitud tomará más tiempo en ejecutarse pero dará mejores resultados en algunos casos. La Rígida se ejecutará más rápido pero en algunos casos no muestra buenos resultados, sin embargo puede ser suficiente para algunas imágenes.

- El interpolador usado afectará la confiabilidad del algoritmo. Se recomienda usar interpolador cúbico.

En la tabla 3.6, se observa la variación del tiempo de cómputo usando el sistema de Neuronavegación usado por el grupo Applied Neuroscience, para los sets 1,2,3. El equipo donde está instalado el sistema es un Intel Core Vpro I7-4940MX 3.10GHz con 4 procesadores físicos y 8 hilos, con 32Gb de ram a 1600MHz. Para el set 1 se obtuvo un promedio de 9919,7228 ms \pm 1282,2815 ms, para el set 2 7596,6261 ms \pm 332,9386 ms y el 3 8577.8095 ms \pm 860,8870 ms. Comparado con los tiempos obtenidos en las pruebas anteriores, para el set1 se redujo en 38 %, para el 2 se redujo un 54 % y para el 3 se redujo 61 %. Se debe recalcar que no se está utilizando todos los núcleos del procesador. Las características del sistema usado para las pruebas se pueden observar en la sección 3.1.

	set1	set2	set3
	Tiempo(ms)	Tiempo(ms)	Tiempo(ms)
1	5769,2753	8538,1968	7991,1643
2	12705,5544	6433,9151	11674,8937
3	16730,4202	7756,7300	7299,0187
4	6457,8252	7091,1402	11312,6201
5	6842,4071	7125,2952	10034,9804
6	8022,7322	8457,1008	3712,2297
7	8433,4268	9789,6632	9343,2855
8	16745,2639	7131,2671	11496,7102
9	8744,1414	6456,4611	5087,5540
10	8746,1823	7186,492	7825,6389
Promedio	9919,7228	7596,6261	8577,8095

Cuadro 3.6: tiempo de cómputo, sets 1,2,3 en el Neuronavegador

Como se aprecia en los resultados anteriores, el tiempo de procesamiento y confiabilidad variarán dependiendo de las imágenes a registrar, por lo tanto es imposible un estimado global del algoritmo.

3.1.3. Robustez

Como se vio en el capítulo 1 y el 2, una de las principales fallas de la IM (normalizada y no normalizada) fue su susceptibilidad al ruido, y en menor proporción a los cambios de iluminación y contraste. Para cuantificar los cambios producidos con estos elementos se preparó un set de imágenes tanto monomodal como multimodal, las cuales se rotaron 10 grados para solo poder analizar un grado de libertad y simplificar las pruebas. Se indujo cambios en las imágenes para observar la variación en la IM y la IMN como se observa a continuación.

Variación de la IM respecto al ruido

En el capítulo 2 se observó una prueba inicial donde se compara una imagen con ruido y 3 tipos de ruido ver 2.1, esa prueba indicaba que la IM tanto normalizada como no normalizada daban un error en la IM por encima del 30 %, con respecto al valor de una imagen sin ruido e incluso no tuviera éxito en el registro. A continuación se observa como cambia la IM con respecto

al ruido Gaussiano, el ruido sal y pimienta tuvo poca variación así que no será incluido. Imágenes analizadas del set 3.

La variación de la IM y la IMN ante ruido Gaussiano con varianza=0.01 ,0.02,0.03 aplicada en una imagen monomodal rotada 10 grados, se observa en las imágenes 3.9a y 3.9b.

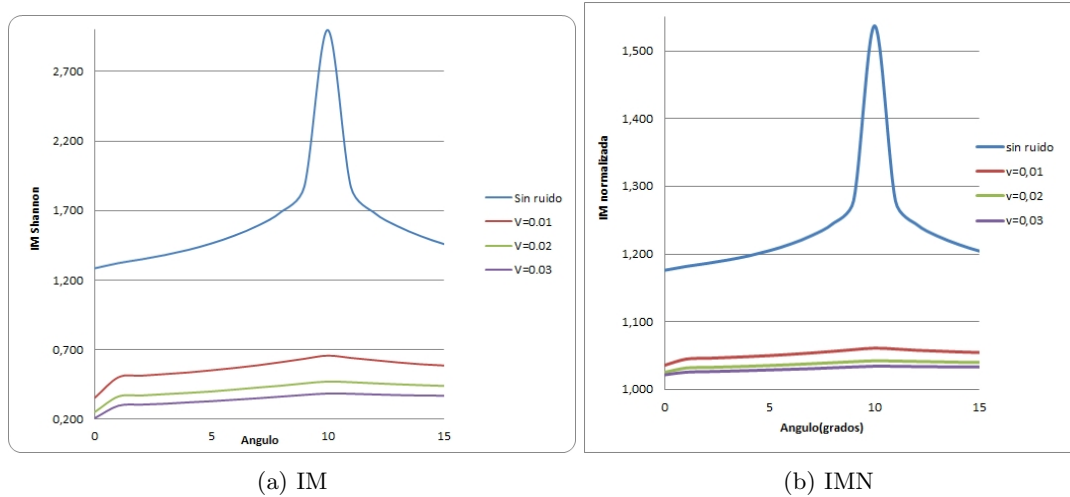


Figura 3.9: Variación de IM (a) y la IMN(b), con imagen monomodal respecto al ruido gaussiano con varianza=0.01 ,0.02,0.03.

monomodal	IM	Error(%)
sinruido	2,9994	0
v=0,01	0,656	78,13
v=0,02	0,4695	84,35
v=0,03	0,3834	87,22
monomodal	IMN	Error(%)
sinruido	1,5373	0
v=0,01	1,0614	30,96
v=0,02	1,0427	32,17
v=0,03	1,0347	32,69

Cuadro 3.7: Cambio de la IM y la IMN respecto al ruido

En la tabla 3.7 se observa que la IM obtuvo una diferencia entre el valor sin ruido y las variaciones con ruido de entre 78 % a 87 %. Como se ve en la sección la IM es muy susceptible a este tipo de ruido. Con la IMN, el error se vio reducido de un 30 % a un 33 %, en este caso la IMN resulta ser mas robusta que la IM. Se recomienda usar una etapa de filtrado, aunque con el uso de la IMN mejora la respuesta ante el ruido gaussiano. Con ruido Poisson, tanto la IM como la IMN fallaron.

Los resultados para una imagen multimodal con ruido Gaussiano, rotada 10 grados, usando IM y IMN se pueden observar en las figuras 3.10a y 3.10b. En la tabla 3.8, se aprecia el error porcentual entre la imagen sin ruido y las imágenes con ruidos gaussianos tanto para la IM como para la IMN. La IMN resulta tener mejor respuesta ante ruido gaussiano que la IM.

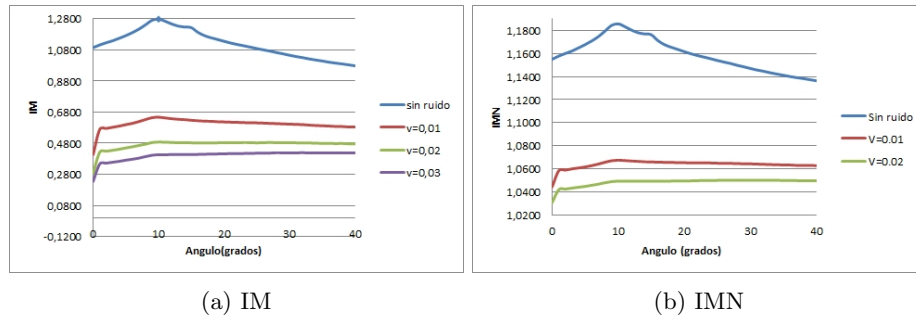


Figura 3.10: Variación de IM (a) y la IMN(b) con imagen multimodal respecto al ruido gaussiano con varianza=0.01 ,0.02,0.03. en $v=0.03$ la IMN falla

MI ruido	IM	Error (%)
sin ruido	1,2783	0,00
$v=0,01$	0,6448	49,56
$v=0,02$	0,4879	61,83
MIN ruido	IM N	Error (%)
sin ruido	1,1857	0,00
$v=0,01$	1,0675	9,97
$v=0,02$	1,0494	11,50

Cuadro 3.8: Cambio de la IM y la IMN respecto al ruido en una imagen multimodal

El comportamiento de la IM y la IMN con respecto a la reducción de ruido se puede observar en las figuras 3.11a y 3.11b. Los reductores de ruido usado fueron un filtro de Wiener adaptativo con ventanas de 3x3 y 5x5, y un filtro de mediana con ventanas de 3x3 y 5x5. En la tabla 3.9 se observa que en este caso la IM como la IMN aumentan cuando se aplican reductores de ruido.

	IMN	Error Porcentual	IM	Error porcentual
sin filtro	1,1833	0,00	1,1857	0,00
Median 3x3	1,1989	1,32	1,3449	13,43
Median 5x5	1,2101	2,26	1,1989	1,11
Wiener 3x3	1,1971	1,17	1,3935	17,53
Wiener 5x5	1,2067	1,98	1,3538	14,18

Cuadro 3.9: Cambio de la IM y la IMN respecto a los reductores de ruido

Variación de la IM respecto a la iluminación y el contraste

En 2.1 se comparó la IM y la IMN con respecto a la iluminación y el contraste de la escena. En dicha prueba indicaba que la variación de la IM y la IMN respecto a estos era del orden del 5 % respecto al brillo y del 10 % al 18 % respecto al contraste. En las pruebas se muestra una variación del factor gamma y del contraste usando técnicas de ecualización del constaste tales como ecualización de contraste normal HE (histogram equalization), ecualización de contraste adaptativo con

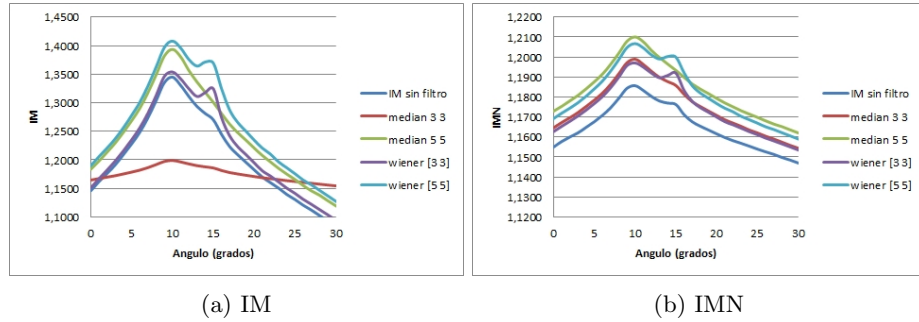


Figura 3.11: Variación de IM (a) y IMN (b) con filtros reductores de ruido

limitador de contraste (Contrast Limited Adaptive Histogram Equalization CLAHE), Brightness Preserving Bi-Histogram equalization BBHE, dualistic sub image histogram equalization DISHE.

La variación del factor Gamma imagen multimodal rotada 10 grados con IM, se puede apreciar en las figuras 3.12a y 3.12b, con un valor gamma de 0,2 la IM falla en el registro, sin embargo para los otros valores no falla en el registro, pero se aprecia una reducción en la IM.

La variación del factor Gamma imagen multimodal rotada 10 grados con IMN se puede apreciar en las figuras 3.13a y 3.13b, con un valor gamma de 0,2 la IMN también falla en el registro, sin embargo para los otros valores no falla en el registro, pero se aprecia una reducción en la IMN.

En la tabla 3.10, se observa el error porcentual entre la IM y la IMN sin corrección gamma y con corrección gamma, tanto la IM como la IMN no tienen éxito en registrar la imagen con gamma 0,2. Se aprecia mejor comportamiento de la IMN respecto a la IM.

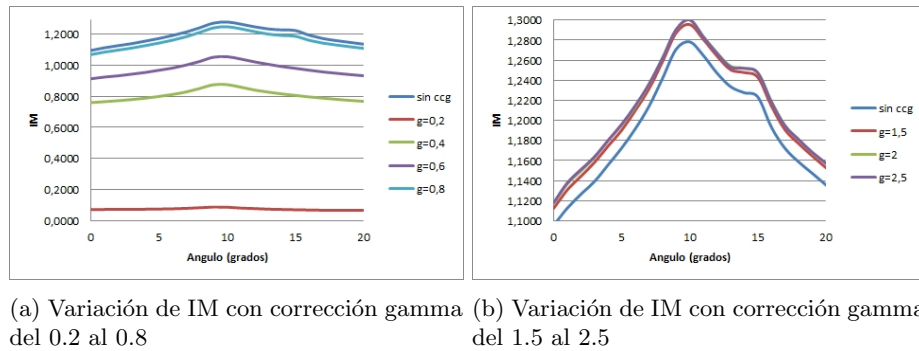
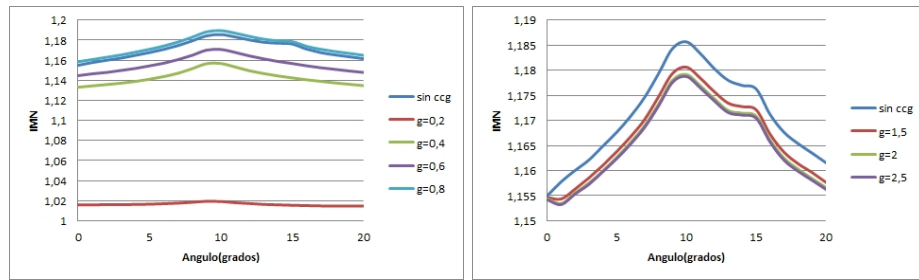


Figura 3.12: Variación de la IM respecto a la corrección gamma

Variación de la IM y la IMN con respecto a los ecualización del contraste: en esta prueba se observó la variación de la IM y la IMN cuando se cambia el contraste de la imagen. El objetivo de la ecualización es hacer mas uniforme el histograma. Entre los métodos usados tenemos ecualización del histograma normal (HE), BBHE , DISHE y CLAHE. En la tabla 3.11 se observa la variación de los métodos, se mide la entropía de las imágenes 1 o 2 para notar los cambios en la información mutua. La entropía de la imagen 1 sin procesar es de 4,2666 y la del imagen 2 es de 3,8958, la información mutua de las imágenes sin procesar es de 1,2783, la información mutua



(a) Variación de IMN con corrección gamma del 0.2 al 0.8 (b) Variación de IMN con corrección gamma del 1.5 al 2.5

Figura 3.13: Variación de la IMN respecto a la corrección gamma

	IMN	Error (%)	éxito	IM	Error (%)	éxito
sin gamma	1,1857	0,00	si	1,2783	0,00	si
0,2	1,0192	14,04	no	0,0869	93,20	no
0,4	1,1568	2,44	si	0,8755	31,51	si
0,6	1,1708	1,26	si	1,0549	17,48	si
0,8	1,1896	0,33	si	1,2479	2,38	si
1,5	1,1806	0,43	si	1,2956	1,35	si
2	1,1792	0,55	si	1,3002	1,71	si
2,5	1,1788	0,58	si	1,3003	1,72	si

Cuadro 3.10: Cambio de la IM y la IMN respecto a la corrección gamma

normalizada es de 1,1857.

	Entropía 1	Entropía 2	IM	Error(%)	IMN	Error(%)
HE	3,0727	3,2044	1,2046	5,77	1,1857	0,19
BBHE	3,9868	3,5179	1,2551	1,81	1,1857	1,11
DISHE	3,8161	3,6657	1,1585	9,37	1,1857	2,61
CLAHE	4,4940	4,4184	1,3017	1,83	1,1857	1,94

Cuadro 3.11: Cambio de la IM y la IMN respecto a la ecualización del contraste usando HE, BBHE, DISHE, CLAHE

3.2. Conclusiones, recomendaciones y trabajos futuros

3.2.1. Conclusiones

- La información mutua demostró ser una herramienta capaz de registrar imágenes multimodales y monomodales, sin embargo algunos artefactos en las imágenes a registrar pueden ocasionar errores en los procesos de registro. Sobre todo en los compuestos por imágenes intraoperatorias donde se logra una confiabilidad del entre el 87 % al 93 % con las imágenes probadas. Pero en muchos otros casos se logra una precisión y confiabilidad cercanos al 100 %.
- La mejor opción en el uso de las métricas fue usar información mutua normalizada, pues se observó un mejor desempeño con respecto a la información mutua no normalizada, como en el efecto del ruido gaussiano o variación del brillo o el contraste. También se apreció una reducción en el tiempo de procesamiento posiblemente porque se eliminan varios máximos locales en la función, con lo que concuerda con lo investigado en la literatura respecto al tema.
- Como consideración de diseño se optó por una transformación rígida (trasladar y rotar) ya que en el proceso de captura del neuronavegador, éste se encargará de escalar las imágenes por medio de un volumen de control por lo tanto no fue necesario. Sin embargo, en las pruebas con diferentes imágenes la transformación rígida resulta insuficiente para el registro de dichas imágenes, por lo tanto se diseñó una etapa de una transformación de similitud (trasladar, rotar y escala), mejorando el registro en ciertas situaciones pero aumentando el tiempo de cómputo. La etapa de transformación de similitud requiere un mejor ajuste de parámetros.
- En la etapa de transformación espacial, el mejor método de interpolación fue el interpolador cúbico, aunque requiere mas tiempo de procesamiento genera menos artefactos indeseables en la imagen que pueden afectar el proceso de registro y generar fallas en este. Sin embargo en varias situaciones consideradas el uso de otros interpoladores resultó óptima y con un menor tiempo de procesamiento.
- El mejor algoritmo encontrado dentro de los analizados fue el PSOAIW dicho algoritmo fue capaz de buscar la mejor solución en varios set analizados con una tasa de acierto cercano al 100 %, sin embargo en algunas imágenes, tubo una tasa de éxito reducida hasta 87 % esto se atribuye a la cantidad de máximos locales que se generan en el espacio de búsqueda por condiciones propias de las imágenes o artefactos inducidos en las etapas de interpolación en las transformaciones espaciales.

3.2.2. Recomendaciones

- El algoritmo diseñado basado en información mutua tiene una gran sensibilidad al ruido sobre todo el ruido gaussiano, por lo tanto es recomendable incluir una etapa de reducción de ruido en las imágenes. Sin embargo, esta etapa puede inducir ciertos artefactos en las imágenes que pueden generar un fallo en el proceso de registro, por consiguiente es necesario modificar los parámetros de la etapa de reducción de ruido a fin de conseguir resultados óptimos.
- En la parte de optimización se varió algunos de los parámetros de los algoritmos analizados con el fin de obtener un tiempo de cómputo reducido. Sin embargo, estos parámetros pueden cambiar según las imágenes a registrar y en algunos casos entregar soluciones erradas, por

lo tanto es necesario identificar los parámetros óptimos según la clase de imágenes usadas. También es necesario un estudio mas detallado de estos algoritmos a fin de identificar todos los parámetros que inciden en su funcionamiento y encontrar la mejor configuración.

- Uno de los principales problemas con el registro son los artefactos inducidos en el proceso de transformación espacial como el tipo de interpolador usado, o bordes irregulares creados en la rotación de la imagen, para reducir su efecto se recomienda usar un tipo de interpolador cúbico, ya que otros tipos pueden inducir errores en la medida. También se puede recortar la imagen alrededor de los bordes para tratar de reducir los efectos del fondo en el registro.

3.2.3. Trabajos futuros

- En el desarrollo de la tesis se consideró el uso de la entropía de Rényi para la información mutua, que según la consulta realizada es mas robusta que la entropía de Shannon, pero se presentaron algunos problemas en la implementación del código debido posiblemente a algunas falla en el diseño del algoritmo. Se requiere un trabajo futuro que considere el uso de la entropía no solo de Rényi sino también otros tipos como la Tsallis y realizar una comparación de dichas técnicas y poder determinar cuál es la mejor opción para la inclusión en registro para la navegación quirúrgica.
- En la parte de optimización se usó una serie de algoritmos como PSO, DE o CS, sin embargo dichos algoritmos presentan fallas con lo cual la confiabilidad y precisión del algoritmo decae considerablemente. Aunque en este proyecto se presento una comparación inicial entre dichos algoritmos es necesario la creación de un algoritmo de optimización mucho mejor para la reducción de errores y así poder dar una precisión y confiabilidad y rapidez propias para un entorno medico.
- En primera instancia y por varios factores explicados en el proyecto se consideró una transformación espacial rígida (traslación y rotación), la escala está considerada en el preproceso de la imagen de manera manual, aunque también se implemento una transformación de similitud. Un trabajo posterior sería la inclusión de una transformación afín en el registro, sin embargo lo ideal sería incluir una transformación no rígida y así poder lidiar con problemas como el brain shift comentado en el proyecto.
- En el proyecto no se incluyó programación en paralelo salvo en pequeñas partes del código, un trabajo posterior sería realizar estos algoritmos usando técnicas de programación en paralelo tanto en CPU como GPU's e incluir librerías como CUDA o OpenCL en su desarrollo.

Apéndice A

Manual Librería algoritmos

A.0.4. Algoritmos Matlab

El algoritmo de registro utilizado en Matlab consta de las siguientes etapas:

```
Leer imagen_referencia, imagen_objetivo
SI tamaño(imagen_referencia )= tamaño(imagen_objetivo )
Leer coordenadas_iniciales(angulo_inicial, x_inicial, y_inicial)

[angulo, x,y , mejor_valor]=Optimizar(-informacion_mutua(imagen_referencia,
    imagen_objetivo, coordenadas_iniciales)

imagen_objetivo=rotar(imagen_objetivo, angulo)
imagen_objetivo=trasladar(imagen_objetivo,x,y)
de lo contrario
Mensaje "no son de igual tamaño"
FIN SI
```

Los algoritmos de optimización siempre buscan el mínimo de las funciones a optimizar, si se desea buscar el máximo, se optimiza el negativo de la función.

La función información mutua es:

```
Leer imagen1, imagen2

Calcular histograma_conjunto(imagen1,imagen2)
Calcular hx=entropia_marginal(imagen1)
Calcular hy=entropia_marginal(imagen1)
Calcular hxy=entropia_conjunta(imagen1)
Retornar IM=hx+hy-hxy
```


Para ser utilizada por el optimizador, se debe tener la siguiente forma:

```
funcion informacion_mutua(angulo , x, y)
Leer imagen1, imagen2
img_movil=imagen2;

img_movil=rotar(imagen2,angulo)
img_movil=transladar(img_movil,x,y)

Calcular histograma_conjunto(imagen1,imagen2)
Calcular hx=entropia_marginal(imagen1)
Calcular hy=entropia_marginal(imagen1)
Calcular hxy=entropia_conjunta(imagen1)
Retornar IM=-(hx+hy-hxy)
```

En cada iteración del optimizador, los parámetros ángulo, x , y cambian buscando la mejor solución que minimize la función.

Las funciones utilizadas en Matlab son las siguientes:

Optimizador

```
[x,eval] = fminsearch(@fun,x0)
```

Donde fun es la función a ser minimizada, x0 es un vector con las posiciones iniciales, x es un vector con las mejores posiciones encontradas y eval es el mínimo valor de la función encontrado. Fminsearch ver <http://es.mathworks.com/help/matlab/ref/fminsearch.html> busca el mínimo de una función multivariable sin restricciones usando un método libre de derivadas . En algunas ocasiones puede manejar discontinuidades, en particular si no ocurren cerca a la solución. La función solo puede manejar soluciones locales, por lo tanto es un método de optimizan local y necesita condiciones iniciales, así que la solución que encuentra solo sera mínimos locales y no globales. Únicamente minimiza funciones reales. Fminsearch usa el método de Nelder Mead como algoritmo de optimización.

Información mutua

```
MI2(image_ref,image_target)
```

Calcula la información mutua de dos imágenes, Imagen de referencia **image_ref** y la imagen objetivo **image_target**. Las dos imágenes deben ser iguales en tamaño, estar en escala de grises y tener una profundidad de los niveles de gris de 8bits (0-255). Las imágenes superiores a 8bits generan una matriz muy grande que en ocasiones desbordan la memoria por lo tanto no se incluyen.

La función calcula tanto la entropía marginal de las imágenes, como su entropía conjunta. Para el calculo de la entropía conjunta es necesario calcular el histograma conjunto de las imágenes.

Histograma conjuntos:

```
h=joint_h(image_1,image_2)
```

El objetivo de este algoritmo es extraer el histograma conjunto de un dos imágenes del mismo tamaño. Se debe tener en cuenta que la imagen deber tener una profundidad de colores de 8 bits en escala de grises, pues es necesario crear una matriz de 256×256 , si se necesita una imagen de mas colores va a necesitar una matriz mucho más grande, por ejemplo una profundidad de colores de 16bits tendrá 65536 colores así que necesitara una matriz de 65536×65536 lo cual daría 4294967296 elementos a 2 byte (int16) por elemento serían 8589934592 byte alrededor de unas 8Gb de memoria, así que podría desbordar la memoria del PC si no tiene suficiente RAM.

Transformación espacial

En matlab se implementó una transformación rígida (traslación y rotación), usando dos funciones:

```
img_movil=trasladar(img, x,y)
```

Aplica una transformación espacial que traslada la imagen img hacia una coordenada x,y .

```
img_movel=imrotate(img, ang,'metodo','crop');
```

Función de matlab <http://es.mathworks.com/help/images/ref/imrotate.html> que rota una imagen img, cierto ángulo ang. Para el uso en el proceso de registro se utilizó el parámetro 'crop', para que la imagen resultante no cambie su tamaño. El parámetro 'metodo' es el interpolador utilizado : 'nearest' (vecinos mas cercano), 'bilinear' (bilineal) y 'bicubic' bicúbica.

A.0.5. Funciones adicionales

Algunas funciones para recortar imágenes:

```
imcortar(im,x,y,tam)  
imcortar_cir(im,x,y,tam)
```

La función `imcortar` recorta un cuadrado en el punto x,y de tamaño tam. La función `imcortar_cir` recorta un círculo en el punto x,y de tamaño tam.

Algoritmos de IM en matlab:

A continuación se dará una version del histograma conjunto y la Información mutua escrita en matlab por Kateryna Artyushkova, ver <http://www.mathworks.com/matlabcentral/>

fileexchange/4534-automatic-image-registration, con algunas modificaciones. Estas funciones sirvieron de base para el algoritmo de registro implementado.

```
function h=joint_h(image_1,image_2)

%
%calcula el histograma conjunto de dos imágenes de igual tamaño
% y de profundidades de colores 255
%
% escrito por:  Kateryna Artyushkova
% Postdoctoral Scientist
% Department of Chemical and Nuclear Engineering
% The University of New Mexico

%image_1=im2uint8(image1);
%image_2=im2uint8(image2);
rows=size(image_1,1);
cols=size(image_1,2);
N=256;

h=zeros(N,N);

for i=1:rows;    % col
    for j=1:cols; % rows
        h(image_1(i,j)+1,image_2(i,j)+1)= h(image_1(i,j)+1,image_2(i,j)+1)+1;
    end
end
end
```

Para la IM no normalizada

```
function h=MI2(image_1,image_2,method)

%
%calcula la Información mutua de dos imágenes de igual tamaño
% escrito por:  Kateryna Artyushkova
% Postdoctoral Scientist
% Department of Chemical and Nuclear Engineering
% The University of New Mexico

a=joint_h(image_1,image_2); % calcula el histograma conjunto
[r,c] = size(a);
[r1,c1]= size(image_1)
b= a./(r1*c1); % histograma conjunto normalizado
y_marg=sum(b); % suma de filas las probabilidades del histograma
%conjunto equivalente al histograma
%de probabilidades de la imagen 1
x_marg=sum(b');%suma de las columnas equivalente al histograma
%de probabilidades de la imagen 2
```

```

Hy=0;
for i=1:c;    % col
    if( y_marg(i)==0 )
        %do nothing
    else
        Hy = Hy + -(y_marg(i)*(log2(y_marg(i))));
        %entropia marginal de la imagen 1
    end
end

Hx=0;
for i=1:r;    %rows
    if( x_marg(i)==0 )
        %do nothing
    else
        Hx = Hx + -(x_marg(i)*(log2(x_marg(i))));
        %entropia marginal de la imagen 2
    end
end
h_xy = -sum(sum(b.*(log2(b+(b==0))))); %entropia conjunta

h = Hx + Hy - h_xy;% Informacion mutua

end

```

A.0.6. Funciones en el lenguaje CSharp

En esta sección se encuentran los manuales de los códigos desarrollados en C#

Algoritmos de registro

Clase informacionmutua

```
public static float entropia(float[] marg)
```

Calcula la entropia marginal, se le ingresa un vector float que corresponde a la densidad de probabilidades de la imagen, este se calcula a partir de histograma conjunto de las imágenes, devuelve un valor de la entropia de Shannon.

```
public static float[] sumar_vectores_double(float[] matriz, int met, int fil, int col)
```

Suma las filas o las columnas de una matriz, la matriz esta en el formato flatten array ver A.0.6, ósea es un array unidimensional que contiene la información de una matriz (para mejorar la velocidad), int met es 0: filas, 1 columnas , fil = filas de la matriz y col es la columnas de la matriz.

```
public static byte[] bitmaptoarray(Bitmap bitmap)
```

Convierte un tipo bitmap (imagen de c#) a un array de bytes.

```
public static float[] joint_histogram(byte[] im1, byte[] im2, int rows ,int cols)
```

Calcula el histograma conjunto de las imágenes en formato de array de byte[], previamente convertidas usando bitmaptoarray, int rows = ancho de la imagen, int cols = altura de la imagen. Devuelve un array de floats de tamaño 256 x 256.

```
public static float mutual_information_shannon(float[] histograma_conjunto, int x, int y)
```

Entrega la información mutua normalizada a partir del histograma conjunto, int x es el altura y y el ancho.

Transformaciones espaciales

Para una transformación rígida:

```
public static IplImage rigida(IplImage img, double x, double y, double ang)
```

Se usa una imagen IplImage img, donde se aplica una traslación en x,y y se rota un ángulo ang. Se retorna un IplImage.

Para una transformación de similitud:

```
public static IplImage similitud(IplImage img, double x, double y, double ang,double esca)
```

Se usa una imagen IplImage img, donde se aplica una traslación en x,y y se rota un ángulo ang y se escala con esca. Se retorna un IplImage.

Fusión

La fusión de las imágenes registradas se realiza con la función:

```
double alpha = 0.5;
double beta;
beta = (1.0 - alpha);
Cv.AddWeighted(img1, alpha, img2, beta, 0.0, dstimg);
```

Donde img1 y img2 son las imágenes a fusionar, alpha y beta son los pesos relativos que cada imagen tendrá en el momento de la fusión y dstimg es la imagen donde quedara guardada la fusión.

Librería de optimización Liboptimization

.

La parte del proceso de optimización es la clase optimizar declarada :

```
class optimizar : LibOptimization.Optimization.absObjectiveFunction
```

Esta clase se encarga del proceso de optimización, tiene una forma especifica según la librería liboptimization el formato es:

Se definen las variables globales como las imágenes en el formato public static

```
public override double F(List<double> ai_var)

{
// funcion a calcular ai_var contiene los parámetros a optimizar en nuestro caso
double x1 = ai_var[0];
double x2 = ai_var[1];
double x3 = ai_var[2];

// aca se coloca la función a minimizar o maximizar con los parametos x1,x2,x3
}
```

x1= angulo, x2=coordenada x , x3=coordenada y se crea cualquier función que se requiera la función retorna un valor h, para minimizar se calcula +h para maximizar se calcula -h este formato se definen el gradiente y la matriz hessiana, para este caso como no se van a utilizar métodos basados en derivadas y por lo tanto van null.

```
public override List<double> Gradient(List<double> aa)
{
    return null;
}

public override List<List<double>> Hessian(List<double> aa)
{
    return null;
}

//este se requiere para determinar el numero de variables a optimizar
public override int NumberOfVariable
{
```

```

        get
        {
            return 3;
        }
    }
}

```

En la NumberOFVariable se colocan el número de variables que el sistema optimizará; en este caso serán 3 (x,y,ángulo) para transformación rígida y 4 para afín.

forma de invocar la clase para optimizar:

Se cargan las imágenes que están globales en la clase optimizar: ejm, se define la imagen public static en la clase optimizar.

```
public static IplImage mat10 = new IplImage();
```

Donde se llame se puede invocar de la siguiente manera:

```
IplImage mat1 = new IplImage("nombrefoto.jpg", LoadMode.GrayScale);
optimizar.mat10 = mat1;
```

se declara la función:

```
var func = new optimizar();
```

luego se declara el método de optimización entre los cuales se encuentran:

```

var opt = new clsOptCS(func);    // Búsqueda Cuco
var opt = new clsOptDE(func); //Evolución diferencial
var opt = new clsOptFA(func); // Algoritmo Firefly
var opt = new clsOptNelderMead(func, 5, 3, 1e-004, 1, 2, 0.5, 2);
//método nelder Mead
var opt = new clsOptNelderMeadWiki(func, 5, 3, 1e-004, 1, 2, 0.5, 2);
//método nelder Mead ver Wikipedia
var opt = new clsOptNewtonMethod(func, 5, 3, 1e-004, 1);
//método de newton
var opt = new clsOptPatternSearch(func, 5, 10000, 1.e-4, 0.6, 2);
// Búsqueda patrones de hooke y jeeves
var opt = new clsOptPSO(func); // Optimizacion por enjambre
//de partículas básico PSO
var opt = new clsOptPSOAIW(func); // PSO adaptive inertia weight
var opt = new clsOptPSOChaoticIW(func); //PSO chaotic inertia weight
var opt = new clsOptPSOLDIW(func); //PSO linear decrease inertial weight
var opt = new clsOptRealGABLX(func); //algoritmo genético GA
var opt = new clsOptRealGASPX(func); //variante del GA
var opt = new clsOptRealGAREX(func); //variante del GA
var opt = new clsOptRealGAUNDX(func); //variante del GA
var opt = new clsOptSimulatedAnnealing(func); // recocido simulado
var opt = new clsOptSteepestDescent(func, 5, 3, 1e-004, 0.3);
// método del gradiente descendiente

```

Cada tipo de optimizador listado tiene sus propios parámetros que pueden ser diferentes entre si, algunos como PatternSearch los parámetros vienen incluidos en la llamada de la función, entre los más comunes tenemos :

`opt.PARAM_EPS` en algunos casos `opt.EPS`, epsilon es un número positivo que controla la exactitud de la convergencia y controla el tamaño mínimo de los valores en la búsqueda, entre más pequeño el valor el algoritmo convergerá mas lento pero sera mas exacto, si es muy grande sera mas rápido pero menos preciso. Alrededor de $1e-3$ es un valor adecuado para el algoritmo de registro utilizado.

`opt.PARAM_Size` Para PSO ajusta la población inicial. `opt.PARAM_PopulationSize` En algunos métodos es la población inicial de las soluciones. Este valor tendrá diferentes comportamientos según el método usado. `opt.PARAM_MAX_ITERATION` en algunos casos `opt.Iteration` es el número máximo de iteraciones que puede ejecutar el programa.

Para iniciar la optimización se usa el comando `optimizacion opt.Init();`.

Luego `opt.DoIteration();` se encargara del proceso de optimización. En `opt.Result` se guardan los parámetros donde se obtuvo el mejor resultado ejm: `opt.Result[0]` será el resultado de la primera variable, `opt.Result[1]` el resultado de la segunda

Técnicas para la optimización del código

En esta sección se explicará brevemente como se optimizo el código en C#. Al hablar de este tipo de optimización no se esta refiriendo a los procesos de optimización matemática sino ciertos trucos y técnicas que logran que se ejecute mas rápido el código. Muchos de estas técnicas son extraídas de foros o paginas web de entusiastas del tema como en <http://www.dotnetperls.com/flatten-array>. Sin embargo su uso a logrado reducir el tiempo de procesamiento sobre los algoritmos. Entre los mas utilizados tenemos:

Flatten arrays (aplanar arrays): el uso de arrays multidimensionales puede ser lento, como es el caso de las matrices de las imágenes en blanco y negro que son 2D. Se puede crear un array de una dimensión que tenga la información de todo el array 2D. Para eso es necesario usar una aritmética adecuada para su uso.

Creación : un array de dimensiones X, Y (largo X y ancho Y), tendrá $X \times Y$ elementos así que es necesario crear un array 1D de $X \times Y$ elementos.

Notación Array o matriz de 2D con coordenada x, y : `array[x, y]`

Notación Flatten array para acceder a la coordenada x, y : `array[x + y \times ancho]`

Eliminar ciclos en un for : el uso del ciclo for hace que el código se ejecute mucho mas lento. Para solucionar este problema se puede ejecutar varias opciones en el mismo ciclo. Por ejm se desea inicializar un array 1D de 500 elementos:

```
Para i=1 hasta 500
suma[i]=0
fin
```


Utilizando la eliminación de ciclos: en este caso en cada ciclo se llena dos posiciones del array, y por lo tanto el ciclo for se reduce a la mitad

```
Para i=1 hasta 250  
suma[i]=0  
suma[i+1]=0  
fin
```

Se puede realizar para un 4 o 5 veces por ciclo.

Apéndice B

Manuales sobre comunicación DICOM

Para la comunicación con dispositivos tales como angiógrafos o ecógrafos (ultrasonido), equipos de resonancia magnética o tomografía que cumplan con el estándar DICOM, en lo referente a las comunicaciones, se pueden utilizar los siguientes programas y librerías para lograr dicho fin.

B.1. Manual DVTK

DVTK es un proyecto de código abierto para probar, validar y diagnosticar protocolos de comunicación en ambientes médicos. Soporta DICOM, HL7 (Health Level Seven) y IHE (Integrating the Healthcare Enterprise). Aunque está diseñado para pruebas de validación de dispositivos o pruebas para emular algún servidor DICOM así de proveer un modelo de información, con el cual no es necesario montar un server DICOM completo y es suficiente solo con módulos para probar ciertas características. DVTK es un programa solo de Windows no tiene soporte en Linux ni MAC OSX.

Entre los módulos empleados tenemos:

- Query Retrieve SCP Emulator: este emulador crea un modelo de información para la realización de consultas sobre el, con lo cual se puede probar comandos DIMSE tales como C-FIND o C-MOVE. El software soportada 3 modelos de información: Patient level, Study level y Patient-Study level.
- Storage SCP emulator: puede emular la funcionalidad de guardado de una estación de trabajo o de un sistema PACS. Se pueden validar datos, permitir o denegar duplicados y crear directorios DICOM.
- DICOM Network Analyzer: permite capturar fácil y rápidamente paquetes de comunicación DICOM sobre TCP/IP de la red con lo que se puede visualizar los mensajes que producen los comandos DIMSE. Esto Permite analizar fallas o problemas en la conexión o en la validación de protocolos

Instalación: cada módulo se instala por separado, siguiendo las instrucción que aparecen en <http://dicom.dvtk.org/modules/wiwi/mod/index.php?page=Downloads&cmenu=downloads>. Es necesario el DICOM Definition Files como prerequisite para todos los módulos.

B.1.1. Configuración del modulo Query Retrieve SCP Emulator

Configuración: Se necesita crear una base de datos con imágenes DICOM, para ello se necesita copiar algunas imágenes a la carpeta de instalación de la VDTK, por defecto:

C:\Program Files (x86)\DVT\Query Retrieve SCP Emulator\Data\QueryRetrieve

Estas imágenes servirán para emular alguna estación de trabajo donde se guardan los archivos DICOM. Una vez configurado este paso, en QR SCP emulator, en la opción configuración se configura cual es el modelo de información que estará activo:

- Patient Root Information Level
- Study Root Information Level
- Patient Study Root Information Level

Se puede escoger todos o uno específico. Para ver el modelo de información se da click en View information model. Se desplegará la base de datos creada y se podrá observar los niveles específicos de cada modelo y los tags DICOM activos para dicho modelo en específico. Se recuerda que la información creada estará agrupada por modelo específico. Si se escogió por ejemplo patient root las imágenes estarán agrupadas por el nombre del paciente, así sean de diferentes estudios, estas imágenes estarán subdivididas en el nivel Series y image como se ve en la figura B.1.

Luego se escoge un AETitle para el emulador en la opción LocalAETitle, por defecto DVTK_QR_SCP, también para QR SCU por defecto DVTK_QR_SCU, así funcionarán los dos el que provee SCP y el solicitante SCU. Se escoge también un puerto de comunicación 106 o 104 por defecto, pero puede ser cualquiera que no interfiera con otro puertos del sistema como 80 o el 23. El timeout es el tiempo en el cual rechaza la conexión si no tiene éxito. Enable case sensitive Query, se recomienda activar todas. También se puede agregar tags a modelo escogido usando el botón Add attributes to information models. Se puede apreciar la configuración en la figura B.2.

Luego en la pestaña move destinations se agrega los datos en la cual se podrán compartir los datos. Este paso es importante pues como se mencionó antes C-MOVE requiere que el SCP conozca el SCU antes de ejecutar el movimiento de archivos el cual es un paso de seguridad para evitar copias no autorizadas. Solo se agrega el nombre para identificarlo, el AETitle, la dirección IP y el puerto. Si por ejemplo tenemos un servidor de guardado con un AETitle ACME1 hay que agregarlo antes de poder mover los archivos usando C-MOVE.

Cuando ya esté configurado se pone a correr en la opción start, para parar el servidor en la opción stop. Ahora él está escuchando por solicitudes las cuales pueden ser C-FIND, C-MOVE. En la pantalla se podrá ver cuando se detecte alguna conexión. Cuando se para el servicio se puede ver los resultados y diagnosticar algún problema. Una prueba de conexión usando un programa realizado en C# con la librería GDCM se puede ver en la figura B.3

FormInformationModel

Information Model

☒ Patient Root ☐ Study Root ☐ Patient/Study Only Root

Patient Level		Study Level		Series Level		Image Level	
Patient Name	Patient ID	Study ID	Study Instance UID	Series Number	Series Instance UID	Instance Number	SOP Instance UID
TORRES SIABATO "E..."	CC2925161	1	100.118.116.2005.2...				
One "Secondary Captur...	SC-11	1	100.118.116.2005.2...				
Two "Secondary Captur...	SC-12	1	100.118.116.2005.2...				
Three "Secondary Capt...	SC-13						
Anonymized							

Tag	Name	VR	Value
0008,0052	Query/Retrieve Level	CS	PATIENT
0010,0010	Patient's Name	PN	One "Secondary Capture Image
0010,0020	Patient ID	LO	SC-11
0010,0030	Patient's Birth Date	DA	
0010,0040	Patient's Sex	CS	O
0020,1200	Number of Patient related s...	IS	3
0020,1202	Number of Patient related s...	IS	5
0020,1204	Number of Patient related i...	IS	8

Figura B.1: Modelo de informacion de QR SCP emulator

QR SCP Emulator

File | Stored Files | About

Configuration | Move Destinations | Edit DCM Files | Activity Logging | Results

Start

Stop

Specify TS

Save

View information model

Local AE title (QR SCP & Store SCU): DVTK_QR_SCP

Local port: 106

QR SCU AE Title: DVTK_QR_SCU

Timeout: 90 Sec (0 for maximum timeout)

Enable Case sensitive Query

☒ Application Entity (AE) ☒ Code strings (CS) ☒ Short String (SH)

☒ Person name (PN) ☒ Long string (LO)

Query Retrieve Information models

☒ Patient Root Information Model

☒ Study Root Information Model

☒ Patient Study Root Information Model

Import Dicom files... Import DICOM files for emulating QR responses.

☐ Select data directory for sending QR responses

Add Attributes to Information model

Figura B.2: Configuración de QR SCP emulator

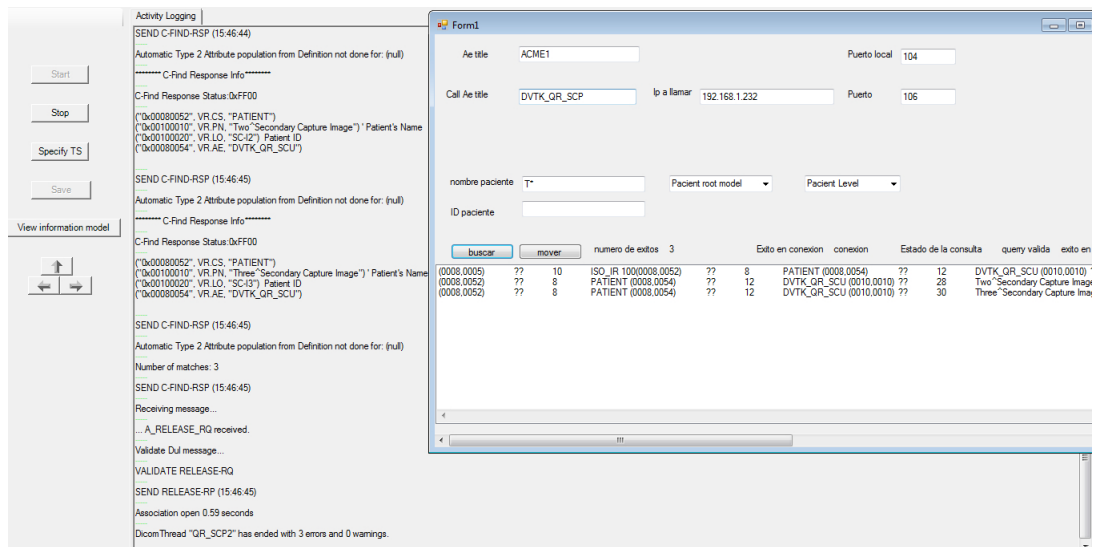


Figura B.3: Prueba de conexión entre GDCM y QR SCP emulator

B.1.2. Configuración del módulo Store SCP emulator

La configuración solo es necesario un Aetitle, un puerto de escucha y ponerlo a ejecutar. Él escuchará ese puerto específico y si recibe algún mensaje de STORE él lo atenderá y en la opción store file se podrá ver los resultados y las imágenes guardadas como se en la figura B.4. Se recuerda que la configuración del servidor tiene que coincidir con la configuración descrita en Query Retrieve SCP Emulator.

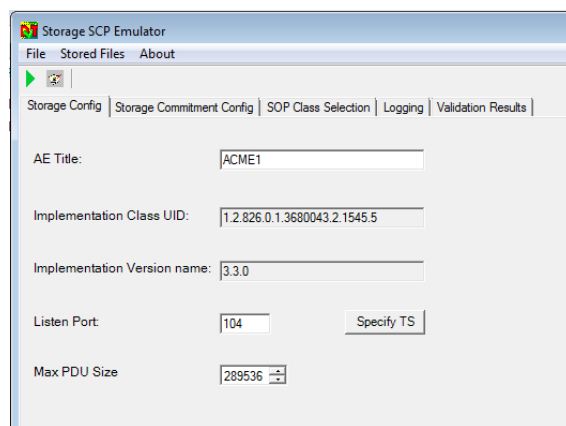


Figura B.4: Configuración del Store SCP emulator

B.1.3. Dicom Network Analyzer

Básicamente el software podrá capturar y analizar los paquetes de red que contengan información DICOM. Al iniciar el servicio capturar cualquier tipo de comunicación DICOM hacia la IP del equipo en cuestión, al detener el servicio se podrá ver un informe de la actividad que ayudará a diagnosticar problemas en la comunicación.

B.2. Manual GDCM

Grassroots DICOM (GDCM) es una librería de código abierto hecha en C++ para la implementación del estándar DICOM. Ofrece un wrapping a varios lenguajes de programación tales como:

- Python
- C #,
- Java
- PHP (experimental)
- Perl (experimental)

Ademas soporta varios codificaciones de imágenes tales como

- RAW
- JPEG lossy 8 y 12 bits (ITU-T T.81, ISO/IEC IS 10918-1)
- JPEG lossless 8-16 bits (ITU-T T.81, ISO/IEC IS 10918-1)
- JPEG 2000 reversible y irreversible (ITU-T T.800, ISO/IEC IS 15444-1)
- RLE
- Deflated (compression at DICOM Dataset level)
- JPEG-LS (ITU-T T.87, ISO/IEC IS 14495-1)

también soporta operaciones de comunicación DICOM SCU de red tales como :

- C-FIND
- C-ECHO
- C-STORE
- C-MOVE

B.2.1. Instalación

La librería GDCM se encuentra en la siguiente ruta <http://sourceforge.net/projects/gdcm/?source=navbar>, donde se puede descargar la ultima versión, 2.6 a la fecha de este texto. Sin embargo se probó con la versión 2.4.0 que trae la librería precompilada http://gdcm.sourceforge.net/wiki/index.php/Installing_GDCM_2.4.0, pero se actualizó en el espacio de elaboración de este texto y es posible que ya sea deprecada la versión 2.4.0. La instalación de la librería precompilada es fácil, sin embargo si se desea compilar alguna versión de GDCM se puede utilizar las siguientes instrucciones descritas en la pagina de GDCM, este procedimiento usa CMAKE para generar los archivos binarios. En linux Ubuntu 14.04 LTS ya tiene lista una versión para instalar desde el centro de control Ubuntu.

En la carpeta de instalación por defecto en Windows es `C:\Program Files (x86)\GDCM 2.4\bin`, donde se encuentran varias aplicaciones para el manejo de archivos DICOM, como los comandos DIMSE C-Find o C-Move. Se pueden encontrar estos comandos con el nombre `gdcm SCU`, este se puede ejecutar en una ventana de comandos en Windows, o si se instaló en linux en un terminal.

B.2.2. Programa gdcm SCU

El programa `gdcm SCU` contiene los comandos DIMSE: C-ECHO, C-FIND, C-MOVE, C-STORE. Un tutorial está descrito en la pagina oficial. A continuación se verán algunos ejemplos básicos ilustrativos:

la sintaxis usada es:

```
gdcm SCU [OPTION]...[OPERATION]...HOSTNAME...[PORT]...
```

Opciones:

```
-H --hostname %s Nombre del host.
-p --port %d Numero del puerto.
--aetitle %s AE Title usado.
--call %s AE Title llamado.
```

Las opciones de modo:

```
--echo C-ECHO (por defecto si no se especifica).
--store C-STORE.
--find C-FIND.
--move C-MOVE.
```

Opciones de C-STORE:

```
-i --input %s nombre de archivo DICOM
-r --proceso recursivo (subdirectorios)
--store-query %s Almacenamiento de la consulta
```

Opciones de C-Find:

```
--patientroot C-FIND Modelo Patient Root.
--studyroot   C-FIND Modelo Study Root Modelo.
--patient     C-FIND Consulta en información Patient (no puede ser usado con --studyroot).
--study       C-FIND Consulta en información Study.
--series      C-FIND Consulta en información Series.
--image       C-FIND Consulta en información Image.
--key         %d,%d[=%s] 0123,4567=VALUE para especifico criterio de búsqueda
              (se puede usar caracteres comodines) también se puede dejar en
              blanco (ie, --key 10,20="" o --key 10,20) para visualización
```

Opciones de C-Move

```
-o --output %s DICOM nombre de archivo / directorio
--port-scp %d Puerto del asociación
--key      %d,%d[=%s] 0123,4567=VALUE para especifico criterio de búsqueda
              (no se aceptan caracteres comodines)
es la misma consulta que C-Find pero sin caracteres comodines
```

Ejemplos :

Sintaxis para un C-Echo simple:

```
gdcmscu --echo dicom.example.com 104
```

El puerto usando es el 104, dicom.example.com puede ser cualquier dirección IP que contenga algún server DICOM. Como se verá más adelante se puede configurar un servidor DICOM sencillo usando DVTK o ORTHANC. En el caso necesario de montar un servidor de prueba se puede usar la IP del equipo por ejemplo “192.168.1.1.104”, si se tiene una IP dinámica se puede averiguarla usando el comando ifconfig en linux o ipconfig en windows. También se puede acceder a servidor dicom de prueba en linea conocido como <http://dicomserver.co.uk/>, se puede utilizar esta dirección web o la IP 213.165.94.158 en los puertos 104 o 11112.

Sintaxis para un C-FIND:

```
gdcmscu --find --patient 213.165.94.158 11112 --patientroot --key 10,10="F*"
```

Este buscará el tag 10,10 osea nombre del paciente “F*”, buscará todos los nombres de paciente que empiecen con F en el servidor <http://dicomserver.co.uk/>. Se usa el modelo de información Patient “Patient root”, en el nivel paciente “patient”.

Sintaxis para un C-MOVE:

Para realizar un C-MOVE es conveniente primero realizar una consulta al servidor DICOM usando C-FIND, por ejemplo se desea comunicar a un servidor con dirección IP 192.168.1.225 a través del puerto 106, usando el modelo de información patientroot, buscando todos los nombres de paciente (tag 0010,0010) que empiecen con T y que muestre la ID del paciente (Tag 10,20):


```
gdcmscu --find --patient 192.168.1.225 106 --patientroot --key 10,10="T*" --key 0010,0020
```

El resultado se aprecia en la figura B.5

Una vez obtenidos algunos datos procedemos a realizar un C-MOVE con la siguiente sintaxis:

```
gdcmscu --move --patient --aetitle ACME1 --call DVTK_QR_SCU --port-scp 11112
192.168.1.225 106--patientroot --key 10,10="Two^Secondary Capture Image "
--key 0010,0020="SC-I2"
```

Ahora es necesario establecer los AETitles de los participantes tanto el que solicita como al que se llama --call estableciendo el puerto del servidor de guardado --port-scp, en este caso 11112 y dando una consulta específica sobre el nombre de paciente (Tag 0010, 0010) y de su identificación ID (tag 10,20), el resultado se aprecia en la figura B.5. Es necesario tener montado algún tipo de servidor DICOM tanto para realizar las consultas como uno para guardar los archivos, esto se puede realizar tanto con la VDTK B.1 como con ORTHANC B.3.

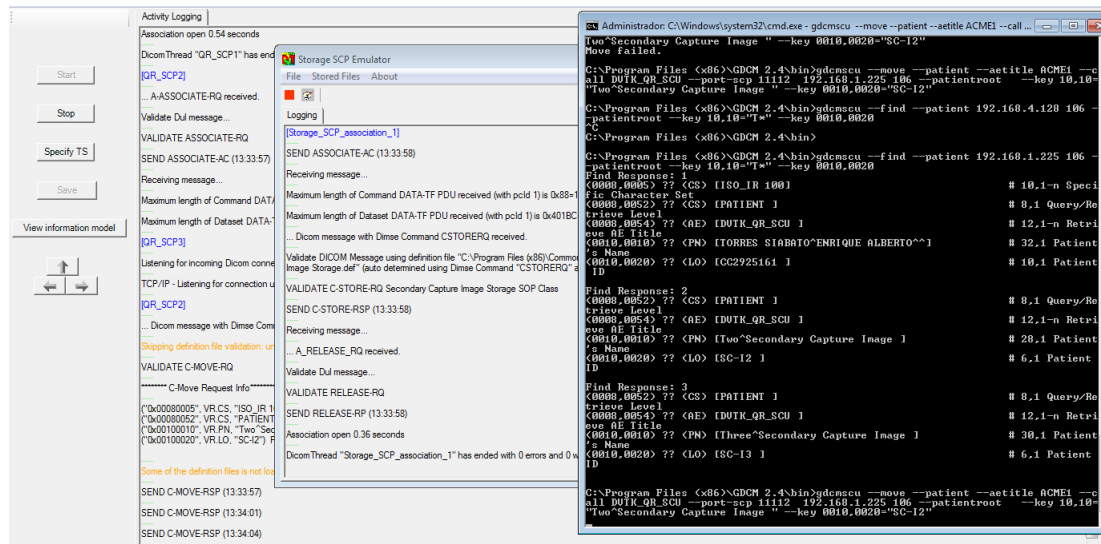


Figura B.5: Prueba del comando C-FIND y C-MOVE

B.2.3. Librería de programación GDCM en Csharp

Una vez realizada la instalación se copian las librerías gdcms-sharp.dll, en el proyecto de C#. Esta librería está ubicada por defecto en Windows C:\Program Files (x86)\GDCM 2.4\lib. En el proyecto de C# se tiene que hacer una referencia a la librería gdcms-sharp.dll, si se está usando por ejemplo Visual Studio en explorador de soluciones, referencia click derecho, agregar referencia.

Una vez referenciado se puede usar en el código fuente:

```
using gdcms;
```

Funciones para los comandos DIMSE en C#

```
C-echo: cecho=CompositeNetworkFunctions.CEcho(ipserver,portserver):
```

Donde ipserver es la dirección ip del destinatario y portserver es el puerto del server. La función retorna true si tiene éxito, false si no lo tiene. La variable cecho que es del tipo bool (true,false) guardará el resultado del C-echo.

C-find :

se necesitan las siguientes variables: un tipo *datasetarrayType* el cual es un vector donde se almacenan los resultados de la búsqueda y se expresa así:

```
DataSetArrayType vector_data_set=new DataSetArrayType();
```

Se necesitan los tags a buscar como en los ejemplos de gdcmscu para esto se utiliza el tipo tag donde se almacenará el tipo de tag a buscar. Ejemplo tag nombre del paciente:

```
gdc.Tag t = new gdc.Tag(0x10,0x10);  
Se pueden agregar otros tag como id del paciente :  
gdc.Tag t2 = new gdc.Tag(0x10, 0x20);
```

Ahora se escoge una variable donde se guarda el tipo de modelo de información que se va a buscar. Ejemplo del modelo Patient (TheRoot), con el nivel patient (TheLevel)

```
gdc.ERootType TheRoot = gdc.ERootType.ePatientRootType;  
gdc.EQueryLevel TheLevel = gdc.EQueryLevel.ePatient;
```

Ahora se crearán unas variables que almacenarán la consulta con el tag y el valor del tag. Primero la variable *KeyValuePairType* que es una variable que contiene dos tipos diferentes de datos. Ejemplo la variable valor1 tendrá (tag,valor_ tag)

```
gdc.KeyValuePairType valor1 = new gdc.KeyValuePairType(t, nombre);  
t es de tipo tag en este caso 10,10 nombre paciente.  
Se puede agregar otra ejemplo  
gdc.KeyValuePairType valor2 = new gdc.KeyValuePairType(t2,nombre) .
```

Luego se crea un array de *KeyValuePairType* llamado *KeyValuePairArrayType*, donde se almacenarán todas las consultas. Ejemplo si se desea almacenar 2 *KeyValuePairType*.

```
gdc.KeyValuePairArrayType clave = new gdc.KeyValuePairArrayType(2);
```

se forma un arrays de vectores pair 2 para incluir a nombre paciente y su id Luego se agrega la variable clave los variables valor1 y valor2

```
clave.Add(valor1);  
clave.Add(valor2);
```

Creación de la consulta: este tipo de dato *BaseRootQuery* tendrá construida la consulta, Ejemplo Buscando el modelo patient, en el nivel patient, las tag (0010, 0010) y (0010,0020) con sus respectivos valores:

```
gdcm.BaseRootQuery theQuery = gdcm.CompositeNetworkFunctions.ConstructQuery
(TheRoot, TheLevel, clave,false);
```

El último campo determina si la consulta es para un c-find (false) o un c-move (true)

Por último se llama el C-find. Ejemplo:

```
exito_buscar = CompositeNetworkFunctions.CFind(ipserver, portserver,
theQuery, vector_data_set ,aetlocal,aetcall);
```

Donde aetlocal es el AETitle del que llama y AETcall, es el AETitle al que se llama. La variable vector_ data_ set almacenará los resultados de la consulta y exito_ buscar que es del tipo bool, será true si tiene éxito y false si falla.

Se puede acceder a los datos vector_ data_ set como un vector ordinario, ejemplo si se tienen 3 éxitos se puede acceder al elemento 2 como `vector_data_set[2].toString()`;

C-Move: En esencia es la misma estructura que C-find, con la diferencia que:

```
gdcm.BaseRootQuery theQuery = gdcm.CompositeNetworkFunctions.ConstructQuery
(TheRoot, TheLevel, clave,true);
```

Va con true en vez de false.

Comando C-move:

```
CompositeNetworkFunctions.CMove(ipserver, portserver, theQuery,
portlocal,aetlocal,aetcall );
```

Es aconsejable primero consultar con C-find, pues C-move no acepta caracteres comodines como * por lo tanto se necesita que la consulta sea lo más explícita posible, incluso se necesitan dos tag como mínimo para su funcionamiento. Una implementación de los comandos C-FIND se puede observar en la imagen B.6

B.3. Manual servidor DICOM: ORTHANC

Orthanc es un servidor DICOM de código abierto que provee un simple pero a la vez poderoso servidor DICOM. Esta diseñado para mejorar el flujo de información DICOM en hospitales y centros de investigación. Orthanc puede convertir sistemas Windows, Linux y OS X en un servidor de almacenamiento DICOM (en esencia un sistema mini-PACS). También Su arquitectura es ligera

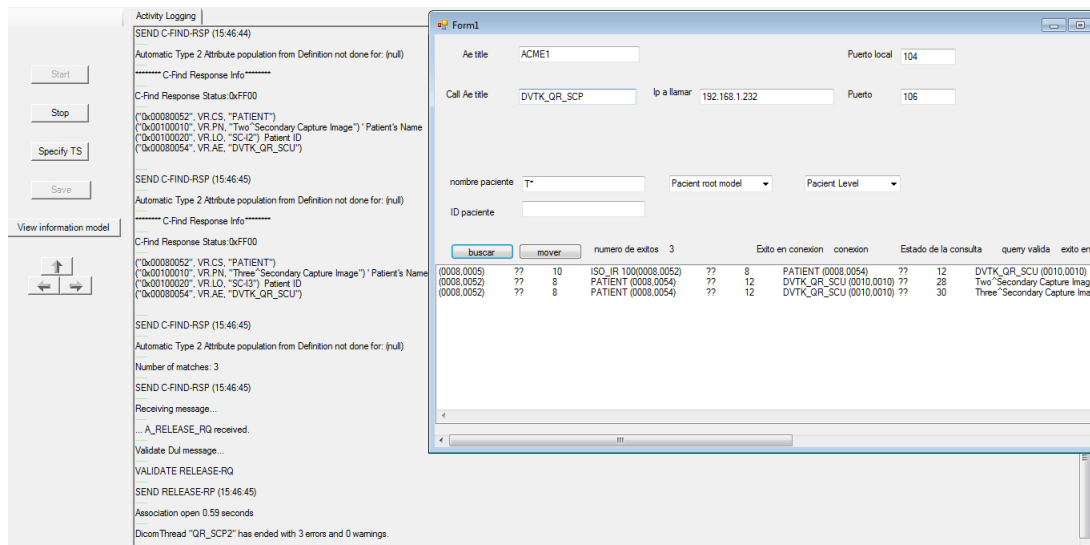


Figura B.6: Aplicacion de prueba creada en Csharp

e independiente y no requiere una complicada administración de bases de datos o dependencias de otros software. Otro factor interesante es que Orthanc es una RESTful API, es decir que su arquitectura utiliza directamente protocolos HTTP para obtener datos o indicar la ejecución de operaciones sobre los datos, en cualquier formato (XML, JSON, etc) sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes, esto hace que Orthanc pueda almacenar información DICOM en un formato JSON que es más estándar entre otros servidores. Estas características lo hacen ideal para el desarrollo de la comunicación DICOM necesaria sin muchas complicaciones ni conocimientos sobre redes.

ORTHANC se puede conseguir este DICOM server de <http://www.orthanc-server.com>, tanto en Linux como en Windows. En la instalación se definen dos carpetas, una la de instalación y otra de almacenamiento, en la última queda un archivo llamado Configuration.json. En Linux ubuntu se puede instalar con:

```
sudo apt-get install orthanc
```

La carpeta de almacenamiento queda en la carpeta del usuario tanto como su archivo de configuración. Una vez instalado ORTHANC queda como un servicio en Windows viendo los servicios locales localizados en el panel de control como se ve en la figura B.7

Se puede iniciar, parar o reiniciar haciendo click derecho sobre el. En Linux podemos reiniciar el servicio con:

```
sudo service orthanc restart
```

También se puede parar: stop, iniciar: start o ver el estado: state.

Se puede iniciar el servidor web como se ve en la figura B.8, con el cual se puede acceder a

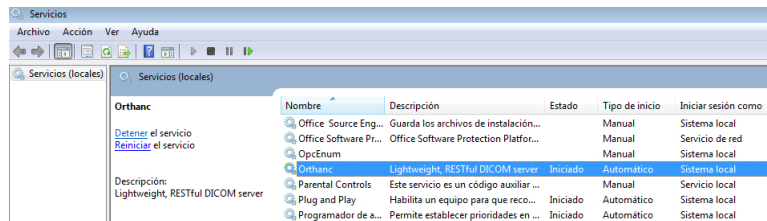


Figura B.7: Servicio Orthanc en windows

servidor DICOM. Esto funciona en cualquier navegador en la dirección <http://localhost:8042> por defecto, el número 8042 es el puerto para este servidor, se puede cambiar igualmente dicho puerto.

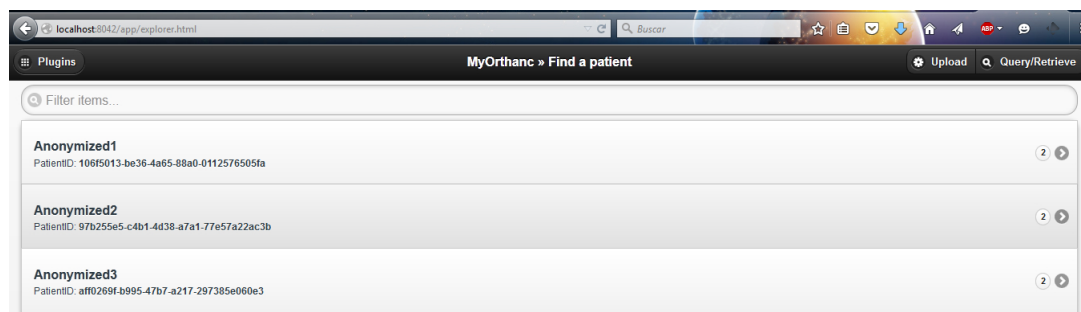


Figura B.8: Servicio WEB Orthanc en windows

Para crear la base de datos con estudios de imágenes médicas tipo DICOM, se da click en Upload de ahí se puede arrastrar desde algún explorador de archivos hacia el navegador. Con esta acción se crea el modelo de información.

Configurar ORTHANC: para esta configuración es necesario editar en algún procesador de texto el archivo Configuration.json ubicada en la carpeta de almacenamiento de ORTHANC definida en la instalación.

En el archivo de configuración se puede definir tanto el AETitle como el puerto y otras opciones. En dicho archivo se busca las siguientes líneas:

```
// The DICOM Application Entity Title
"DicomAet" : "ORTHANC",
// The DICOM port
"DicomPort" : 104,
```

También se definen los otros servicios (Peer) que van a acceder al servidor. Este paso es importante para poder realizar un C-Move. Por ejemplo se desea agregar el QR emulador de la VDTK descrito en, con AETitle : DVTK_QR_SCP, puerto de comunicación 106 y dirección IP 192.168.4.5 se busca la parte en el archivo con:

```
// The list of the known DICOM modalities
```

```
"DicomModalities" : {
```

Y se agrega este:

```
"Dvtk": [ "DVTK_QR_SCP", "192.168.4.5", 106 ],
```

Si se desea agregar otro se coloca(.). Por ejemplo el servidor <http://dicomserver.co.uk/>

```
"DicomServerUK": [ "DicomServerUK", "213.165.94.158", 104 ],
```

Con este paso realizado ya se puede hacer una consulta (C-FIND), por ejemplo a <http://dicomserver.co.uk/> con la opción Query/Retrieve como se ve en la figura, también se puede comprobar el C-ECHO. Dado esta búsqueda se puede pedir un C-MOVE sobre determinado estudio.

Nota el servidor <http://dicomserver.co.uk/> en ocasiones no permite el uso de C-Move, así que esta parte es mejor probarla con otro dispositivo o servidor como DVTK.

The screenshot shows a web browser window with the URL `localhost:8042/app/explorer.html#query-retrieve`. The page title is "MyOrthanc » DICOM Query/Retrieve (1/3)". The interface includes the following elements:

- DICOM server:** A dropdown menu with "Dvtk" selected.
- Field of interest:** A list of radio buttons with "Patient Name" selected. Other options are "Patient ID", "Accession Number", and "Study Description".
- Value for this field:** A text input field containing "f*".
- Study date:** A dropdown menu with "Any date" selected.
- Modalities:** A row of buttons for "CR", "CT", "MR", "NM", "PT", "US", and "XA".
- Buttons:** At the bottom, there are two buttons: "Test Echo" (dark grey) and "Search studies" (blue).

Figura B.9: Query/Retrieve (C-FIND)

Un tutorial mas extenso hecho para sistemas Windows se encuentra en <http://www.orthanc-server.com/resources/2015-02-09-emsy-tutorial/index.html>, escrito por Emsy Chan.

Bibliografía

- [1] Philippe Merloz, Stephane Lavallee, Jerome Tonetti, and Laurence Pittet. Image-guided spinal surgery: Technology, operative technique, and clinical practice. *Operative Techniques in Orthopaedics*, 10(1):56 – 63, 2000.
- [2] P.J. Slotty, M.A. Kamp, C. Wille, T.M. Kinf, H.J. Steiger, and J. Vesper. The impact of brain shift in deep brain stimulation surgery: observation and obviation. *Acta Neurochirurgica*, 154(11):2063–2068, 2012.
- [3] Organizacion Mundial de la salud OMS. ¿qué son los trastornos neurológicos? disponible en internet en 2015: <http://www.who.int/features/qa/55/es>, 2014.
- [4] José Manoel Bertolote. Los trastornos neurológicos afectan a millones de personas en todo el mundo: informe de la oms. disponible en internet en 2015: <http://www.who.int/mediacentre/news/releases/2007/pr04/es/>, 2007.
- [5] Autores varios. *Neurological Disorders: Public Health Challenges*. Organizacion Mundial de la salud, 2006.
- [6] Autores varios. *Atlas: Country resources for neurological disorders*. Organizacion Mundial de la salud, 2004.
- [7] Xiu Ying Wang, Stefan Eberl, Michael Fulham, Seu Som, and David Dagan Feng. *8 - Data Registration and Fusion*. Biomedical Engineering. Academic Press, Burlington, 2008.
- [8] Stefan Klein and Marius Staring. *Elastix the manual*. University Medical Center Utrecht, 2012.
- [9] A. Ardeshir Goshtasby. *2D and 3-D Image Registration: For Medical, Remote Sensing, and Industrial Applications*. Wiley-Interscience, 2005.
- [10] G.P. Penney, J. Weese, J.A. Little, P. Desmedt, D.L.G. Hill, and D.J. Hawkes. A comparison of similarity measures for use in 2-d-3-d medical image registration. *Medical Imaging, IEEE Transactions on*, 17(4):586–595, Aug 1998.
- [11] Derek L.G. Hill and David J. Hawkes. *Handbook of Medical Image Processing and Analysis Chapter 37 - Across-Modality Registration Using Intensity-Based Cost Functions*. Academic Press, Burlington, second edition edition, 2009.
- [12] A. Ardeshir Goshtasby. *Image Registration - Principles, Tools and Methods*. Advances in Computer Vision and Pattern Recognition. Springer, 2012.

- [13] Robert M. Gray. *Entropy and Information Theory*. Springer-Verlag New York., New York, NY, USA, 2011.
- [14] Mohanalin, Beenamol, Prem Kumar Kalra, and Nirmal Kumar. An automatic image registration scheme using tsallis entropy. *Biomedical Signal Processing and Control*, 5(4):328 – 335, 2010.
- [15] Màrius Vila, Anton Bardera, Miquel Feixas, and Mateu Sbert. Tsallis mutual information for document classification. *Entropy*, 13(9):1694, 2011.
- [16] Roger P. Woods. *Handbook of Medical Image Processing and Analysis Chapter 32 - Spatial Transformation Models*. Academic Press, Burlington, second edition, 2009.
- [17] Oscar Andrés Vélez Martínez. *Metodología para el registro multimodal de imágenes 3D utilizando información mutua*. Universidad Tecnologica de Pereira, 2014.
- [18] J. Vandemeulebroucke, E. Vansteenkiste, and W. Philips. *A multi-modal 2D/3D registration scheme for preterm brain images*. Aug 2006.
- [19] Sima Noghanian, Abas Sabouni, Travis Desell, and Ali Ashtari. *Microwave Tomography: Global Optimization, Parallelization and Performance Evaluation*. Springer Publishing Company, Incorporated, 2014.
- [20] Xin-She Yang and S. Deb. Cuckoo search via levy flights. In *Nature Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, pages 210–214, Dec 2009.
- [21] H.S. Lope and L.S. Coelho. Particle swarn optimization with fast local search for the blind traveling salesman problem. In *Hybrid Intelligent Systems, 2005. HIS '05. Fifth International Conference on*, pages 245–250, Nov 2005.
- [22] Lin Lu, Qi Luo, Jun yong Liu, and Chuan Long. An improved particle swarm optimization algorithm. In *Granular Computing, 2008. GrC 2008. IEEE International Conference on*, pages 486–490, Aug 2008.
- [23] Hashmi Adil, Goel Nishant, Goel Shruti, and Gupta Divya. Firefly algorithm for unconstrained optimization. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 11(1):75–78, 2013.
- [24] Xin-She Yang. Firefly algorithm, lévy flights and global optimization. In Max Bramer, Richard Ellis, and Miltos Petridis, editors, *Research and Development in Intelligent Systems XXVI*, pages 209–218. Springer London, 2010.
- [25] Nikolaos Mitianoudis and Tania Stathaki. Pixel-based and region-based image fusion schemes using {ICA} bases. *Information Fusion*, 8(2):131 – 142, 2007. Special Issue on Image Fusion: Advances in the State of the Art.
- [26] Roger P. Woods. *Handbook of Medical Image Processing and Analysis Chapter 33 - Validation of Registration Accuracy*. Academic Press, Burlington, second edition, 2009.
- [27] Gengsheng Lawrence Zeng. *Medical Image Reconstruction A Conceptual Tutorial Methodologies*. Springer-Verlag New York., 2010.
- [28] Jerrold T. Bushberg. *The Essential physics of medical imaging*. Lippincott Williams & Wilkins, 2002.
- [29] Elisa E. Konofagou. *BIOMEDICAL TECHNOLOGY and DEVICES H A N D B O O K Chapter 9. Ultrasonic Imaging*. CRC Press., 2004.

- [30] Walter Serna Serna y Juan Pablo Trujillo Lemus. *Sistema de neuronavegación para ubicación de coordenadas cerebrales mediante técnicas de visión estéreo*. Universidad Tecnológica de Pereira, 2014.
- [31] Medical Imaging and Technology Alliance. *The DICOM Standard 2015*. NEMA, 2015.
- [32] Oleg S. Pinykh. *Digital Imaging and Communications in Medicine: A Practical Introduction and Survival Guide*. Springer Publishing Company, Incorporated, 2 edition, 2012.
- [33] Luis Lanca and Augusto Silva. *Digital Imaging Systems for Plain Radiography*. Springer-Verlag New York, 1 edition, 2013.
- [34] Lorenzo Faggioni, Emanuele Neri, Carlo Castellana, Davide Caramella, and Carlo Bartolozzi. The future of {PACS} in healthcare enterprises. *European Journal of Radiology*, 78(2):253 – 258, 2011. From {PACS} to the clouds.
- [35] R.Suganya, K.Priyadharsini, and S.Rajaram. Article: Intensity based image registration by maximization of mutual information. *International Journal of Computer Applications*, 1(20):1–5, February 2010. Published By Foundation of Computer Science.
- [36] S. Winter, B. Brendel, I. Pechlivanis, K. Schmieder, and C. Igel. Registration of ct and intra-operative 3-d ultrasound images of the spine using evolutionary and gradient-based methods. *Evolutionary Computation, IEEE Transactions on*, 12(3):284–296, June 2008.
- [37] Bernhard Fuerst, Wolfgang Wein, Markus Müller, and Nassir Navab. Automatic ultrasound–mri registration for neurosurgery using the 2d and 3d {LC2} metric. *Medical Image Analysis*, 18(8):1312 – 1319, 2014. Special Issue on the 2013 Conference on Medical Image Computing and Computer Assisted Intervention.
- [38] Sheng Luan, Tianmiao Wang, Weishi Li, Zhongjun Liu, Liang Jiang, and Lei Hu. 3d navigation and monitoring for spinal milling operation based on registration between multiplanar fluoroscopy and {CT} images. *Computer Methods and Programs in Biomedicine*, 108(1):151 – 157, 2012.
- [39] Y. Yamamura, Hyoungeop Kim, Joo Kooi Tan, S. Ishikawa, and A. Yamamoto. An image registration method for head cta and mra images using mutual information on volumes of interest. In *Soft Computing and Intelligent Systems (SCIS), 2014 Joint 7th International Conference on and Advanced Intelligent Systems (ISIS), 15th International Symposium on*, pages 1490–1493, Dec 2014.
- [40] Yipeng Hu, Hashim Uddin Ahmed, Zeike Taylor, Clare Allen, Mark Emberton, David Hawkes, and Dean Barratt. {MR} to ultrasound registration for image-guided prostate interventions. *Medical Image Analysis*, 16(3):687 – 703, 2012. Computer Assisted Interventions.
- [41] Andrew Lang, Parvin Mousavi, Sean Gill, Gabor Fichtinger, and Purang Abolmaesumi. Multi-modal registration of speckle-tracked freehand 3d ultrasound to {CT} in the lumbar spine. *Medical Image Analysis*, 16(3):675 – 686, 2012. Computer Assisted Interventions.
- [42] Chen-Rui Chou, Brandon Frederick, Gig Mageras, Sha Chang, and Stephen Pizer. 2d/3d image registration using regression learning. *Computer Vision and Image Understanding*, 117(9):1095 – 1106, 2013.
- [43] A. Varnavas, T. Carrell, and G. Penney. Increasing the automation of a 2d-3d registration system. *Medical Imaging, IEEE Transactions on*, 32(2):387–399, Feb 2013.

- [44] Robert J. Schneider, Douglas P. Perrin, Nikolay V. Vasilyev, Gerald R. Marx, Pedro J. del Nido, and Robert D. Howe. Real-time image-based rigid registration of three-dimensional ultrasound. *Medical Image Analysis*, 16(2):402 – 414, 2012.